ซอฟต์แวร์สำหรับการแก้ปัญหากำหนดการไม่เชิงเส้นภายใต้เงื่อนไขบังคับสมการเชิงเส้นเป็นช่วง และอสมการไม่เชิงเส้นโดยวิธีระนาบตัด

นางสาว ศุภิสรา ศรีขวานทอง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคณนา ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2547

ISBN: 974-17-6027-2

SOFTWARE FOR SOLVING A NONLINEAR PROGRAMMING PROBLEM WITH EQUALITY PIECEWISE LINEAR CONSTRAINTS AND INEQUALITY NONLINEAR CONSTRAINTS BASED ON A METHOD OF CUTTING-PLANE

Miss Supissara Srikwanthong

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computational Science
Department of Mathematics
Faculty of Science
Chulalongkorn University
Academic Year 2004
ISBN: 974-17-6027-2

หัวข้อวิทยานิพนธ์	ซอฟต์แวร์สำหรับการแก้ปัญหากำหนดการไม่เชิงเส้นภายใต้เงื่อนไขบังคับสมการเชิงเส้นเป็นช่วงและอสมการไม่เชิงเส้นโดยวิธีระนาบตัด
โดย	นางสาว ศุภิสรา ศรีขวานทอง
สาขาวิชา	วิทยาการคณนา
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ วีรศักดิ์ รัตนสมบูรณ์
อาจารย์ที่ปรึกษาร่วม	ผู้ช่วยศาสตราจารย์ ดร. กรุง สินอภิรมย์สราญ

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

…………………………………………. คณบดีคณะวิทยาศาสตร์
(ศาสตราจารย์ ดร.เปี่ยมศักดิ์ เมนะเศวต)

คณะกรรมการสอบวิทยานิพนธ์

…………………………………………. ประธานกรรมการ
(รองศาสตราจารย์ ดร. วนิดา เหมะกุล)

…………………………………………. อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ วีรศักดิ์ รัตนสมบูรณ์)

…………………………………………. อาจารย์ที่ปรึกษาร่วม
(ผู้ช่วยศาสตราจารย์ ดร. กรุง สินอภิรมย์สราญ)

…………………………………………. กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. พรชัย สาตรวาหา)

ศุภิสรา ศรีขวานทอง: ซอฟต์แวร์สำหรับการแก้ปัญหากำหนดการไม่เชิงเส้นภายใต้เงื่อนไขบังคับสมการเชิงเส้นเป็นช่วงและอสมการไม่เชิงเส้นโดยวิธีระนาบตัด (SOFTWARE FOR SOLVING A NONLINEAR PROGRAMMING PROBLEM WITH EQUALITY PIECEWISE LINEAR CONSTRAINTS AND INEQUALITY NONLINEAR CONSTRAINTS BASED ON A METHOD OF CUTTING-PLANE) อ.ที่ปรึกษา:รองศาสตราจารย์ วีรศักดิ์ รัตนสมบูรณ์ อ. ที่ปรึกษาร่วม: ผู้ช่วยศาสตราจารย์ ดร. กรุง สินอภิรมย์สราญ, จำนวนหน้า 78 หน้า. ISBN: 974-17-6027-2

งานวิจัยนี้เน้นการพัฒนาซอฟต์แวร์แก้ปัญหากำหนดการไม่เชิงเส้นภายใต้เงื่อนไขบังคับที่เป็นสมการเชิงเส้นเป็นช่วงและอสมการไม่เชิงเส้นโดยวิธีระนาบตัด วิธีการนี้เป็นการทำซ้ำเพื่อหาค่ามากสุดของฟังก์ชัน Lagrangian dual ที่สอดคล้องกับปัญหา ในขั้นตอนการหาค่าเกรเดียนต์และเฮสเสียนของฟังก์ชันได้เรียกใช้โปรแกรม ADOL-C ในขั้นตอนการหาผลเฉลยของปัญหากำหนดการเชิงเส้นได้เรียกใช้โปรแกรม GNU Linear Programming Kit และในขั้นตอนการหาผลเฉลยของระบบสมการไม่เชิงเส้นใช้วิธี Newton, วิธี steepest descent และวิธี combination of Newton and steepest descent ผลลัพธ์ที่ได้จะถูกนำมาเปรียบเทียบกับซอฟต์แวร์ GAMS ซึ่งเป็นซอฟต์แวร์ทางการค้าที่ใช้แก้ปัญหาเหมาะที่สุด

ภาควิชา	**คณิตศาสตร์**	ลายมือชื่อนิสิต...
สาขาวิชา	**วิทยาการคณนา**	ลายมือชื่ออาจารย์ที่ปรึกษา................................
ปีการศึกษา	**2547**	ลายมือชื่ออาจารย์ที่ปรึกษาร่วม…..……………………..

4472444323 : MAJOR COMPUTATION SCIENCE

KEYWORD: NONLINEAR PROGRAMMING/ CUTTING-PLANE METHOD

SUPISSARA SRIKWANTHONG: SOFTWARE FOR SOLVING A NONLINEAR PROGRAMMING PROBLEM WITH EQUALITY PIECEWISE LINEAR CONSTRAINTS AND INEQUALITY NONLINEAR CONSTRAINTS BASED ON A METHOD OF CUTTING-PLANE. THESIS ADVISOR: ASSOC. PROF. VIRASAK RATANASOMBOON, THESIS CO-ADVISOR: ASST. PROF. KRUNG SINAPIROMSARAN, Ph.D., 78 pp. ISBN: 974-17-6027-2

This research is concentrating on development software to solve nonlinear programming problem with equality piecewise linear constraints and inequality nonlinear constraints based on a method of the cutting-plane. This iterative method is used to find the maximum for the Lagrangian dual function. The procedure of finding gradient and Hessian of function use the ADOL-C program. GNU Linear Programming Kit is chosen to solve a linear programming. The Newton, steepest descent and combination of Newton and steepest descent method is used for solving system of nonlinear equation. We compare the result with the commercial optimization software GAMS.

Department	**Mathematics**	Student's signature…………………………...
Field of study	**Computational Science**	Advisor's signature…………………………...
Academic year	**2004**	Co-advisor's signature………………………..

กิตติกรรมประกาศ

ผู้วิจัยขอกราบขอบพระคุณ รองศาสตราจารย์ วีรศักดิ์ รัตนสมบูรณ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร. กรุง สินอภิรมย์สราญ อาจารย์ที่ปรึกษาร่วม ที่กรุณาให้คำปรึกษา และข้อเสนอแนะแนวทางในการดำเนินการวิจัย ขอกราบขอบพระคุณรองศาสตราจารย์ ดร. วนิดา เหมะกุล ประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร. พรชัย สาตรวาหา กรรมการสอบวิทยานิพนธ์ ที่กรุณาให้คำปรึกษา คำแนะนำและตรวจแก้วิทยานิพนธ์เล่มนี้ให้สำเร็จลุล่วงได้ด้วยดี

ข้าพเจ้าขอกราบขอบพระคุณบิดา มารดาผู้ซึ่งสนับสนุนและให้กำลังใจข้าพเจ้าตลอดมา ขอขอบคุณหน่วยงานต่าง ๆ ที่ให้ความอนุเคราะห์ข้อมูลได้แก่ สำนักหอสมุดกลาง และห้องสมุดภาควิชาคณิตศาสตร์ และสุดท้ายขอขอบคุณทุกท่านที่ให้ความช่วยเหลือ และกำลังใจข้าพเจ้าในการทำวิทยานิพนธ์นี้

สารบัญ

บทที่ 1
บทนำ

ในปัจจุบันองค์กรต่าง ๆ ได้มีการเพิ่มขนาดขององค์กรเองให้โตขึ้นตามการขยายตัวอย่างรวดเร็วของภาวะเศรษฐกิจ ซึ่งทำให้องค์กรนั้นมีความซับซ้อนมากขึ้น เป็นผลให้ผู้บริหารองค์กรนั้นมีการตัดสินใจที่ซับซ้อนขึ้น การตัดสินใจผิดพลาดของผู้บริหารอาจก่อให้เกิดความเสียหายต่อกิจการขององค์กร ทำให้มีผู้พยายามหาวิธีการอย่างมีหลักการในการแก้ปัญหา โดยกำหนดแนวทางของปัญหา จุดประสงค์ วิธีการดำเนินงาน เพื่อให้การตัดสินใจเกิดความผิดพลาดที่น้อยที่สุด และส่งผลดีต่อองค์กรมากที่สุด วิชาการแขนงหนึ่งได้ถูกพัฒนาขึ้นเพื่อหาหลักการดำเนินงานดังกล่าวคือวิชาที่ว่าด้วย การวิจัยดำเนินงาน

การวิจัยดำเนินงาน เป็นวิชาการแขนงหนึ่งซึ่งเกิดขึ้นระหว่างสงครามโลกครั้งที่ 2 เพื่อใช้บริหารงานทางทหาร เมื่อสงครามสิ้นสุดลงความสำเร็จของทีมงานทางทหารที่ใช้หลักวิจัยดำเนินงานเริ่มถูกนำมาใช้ในวงการอุตสาหกรรม เพื่อนำมาแก้ไขปัญหาที่ซับซ้อนด้านการผลิต การขนส่งและการจัดการ ความนิยมของการวิจัยดำเนินงานจึงมีเพิ่มขึ้น จุดประสงค์หลักในการวิจัยดำเนินงาน คือการหาแนวทางปฏิบัติที่ทำให้ได้ผลลัพธ์เหมาะสมที่สุดภายใต้เงื่อนไขที่กำหนด โดยผลลัพธ์ที่ได้ต้องสอดคล้องกับจุดมุ่งหมายของการแก้ปัญหา

กำหนดการเชิงเส้น เป็นเทคนิคในการแก้ปัญหาทางการวิจัยดำเนินงานที่มีลักษณะความสัมพันธ์ของตัวแปรต่าง ๆ ที่เกี่ยวข้องเป็นฟังก์ชันเชิงเส้น เราสามารถมองผลความก้าวหน้าทางคณิตศาสตร์ ซึ่งมีความคิดริเริ่มมาจากนักคณิตศาสตร์และนักวิทยาศาสตร์หลาย ๆ ท่าน เช่น Von Neumann, J. เริ่มใช้ทฤษฎีสูงสุด-ต่ำสุดในทฤษฎีของเกมในปี 1928 และถูกพัฒนานำไปใช้ในปัญหาทางการขนส่งในปี 1941 เทคนิคดังกล่าวนำไปใช้กับการแก้ปัญหาทางโภชนาการในปี 1945 โดย Stigler, G. จากนั้นในปี 1947 Dantzig, G. B., Marshall, C. W. และเพื่อนร่วมงานในกองทัพอากาศสหรัฐอเมริกาได้ใช้วิธีทางคณิตศาสตร์ และเทคนิคที่เกี่ยวข้องมาแก้ปัญหาทางการวางแผนโครงงานในกองทัพ โดยเริ่มจัดรูปองค์กรทั้งหมดให้มีความสัมพันธ์ทางคณิตศาสตร์เป็นฟังก์ชันเชิงเส้น แล้วใช้วิธีทางคณิตศาสตร์แก้ปัญหานั้น ผลงานที่ปรากฏได้รับการยอมรับเป็นอย่างมาก ทำให้เกิดวิธีการที่เรียกว่า simplex method ซึ่งเป็นเทคนิคที่ใช้สำหรับแก้ปัญหากำหนดการเชิงเส้นที่มีประสิทธิภาพ

ปัญหาการวิจัยดำเนินงานบางปัญหามีลักษณะความสัมพันธ์ของตัวแปรต่าง ๆ เป็นแบบไม่เชิงเส้น พิจารณาตัวอย่างต่อไปนี้ โรงงานแห่งหนึ่งผลิตสินค้า 3 ชนิด โดยที่ปริมาณการขายของสินค้าแต่ละชนิดขึ้นอยู่ราคาของสินค้า ยกเว้นสินค้าชนิดที่ 3 ที่ปริมาณการขายของสินค้าจะขึ้นอยู่กับราคาของสินค้าชนิดที่ 2 และชนิดที่ 3 ด้วย สมมุติให้ x_i คือปริมาณการขายของสินค้าชนิดที่ i โดยที่ $i = 1, 2, 3$ ฝ่ายผลิตควรจะดำเนินการผลิตอย่างไรจึงจะได้ผลกำไรสูงที่สุด เมื่อกำไรทั้งหมดที่ได้จากการผลิตสินค้าทั้ง 3 ชนิดมีความสัมพันธ์ดังนี้

$$f(x_1,x_2,x_3) = 4x_1 - x_1^2 + 9x_2 - x_2^2 + 10x_3 - 2x_3^2 - \frac{1}{2}x_2x_3$$

เราต้องการหาค่า x_1,x_2,x_3 ที่จะทำให้ $f(x_1,x_2,x_3)$ มีค่ามากที่สุด โดยที่การผลิตนั้นใช้ทรัพยากรที่มีอยู่อย่างจำกัดคือ

$$\begin{aligned} 0.2x_1 + 0.4x_2 + 0.1x_3 &\le 1000 \\ 0.4x_1 + 0.2x_2 + 0.1x_3 &\le 2000 \end{aligned}$$

ดังนั้นปัญหาจึงอยู่ในรูปแบบ

$$\begin{array}{ll} \text{maximize} & 4x_1 - x_1^2 + 9x_2 - x_2^2 + 10x_3 - 2x_3^2 - \frac{1}{2}x_2x_3 \\ \text{subject to} & 0.2x_1 + 0.4x_2 + 0.1x_3 \le 1000 \\ & 0.4x_1 + 0.2x_2 + 0.1x_3 \le 2000 \\ & \qquad\qquad x_1, x_2, x_3 \ge 0 \end{array}$$

จะเห็นได้ว่าปัญหาข้างต้นมีความสัมพันธ์ของตัวแปรเป็นแบบไม่เชิงเส้น ดังนั้นเราจะเรียกปัญหานี้ว่า ปัญหากำหนดการไม่เชิงเส้น

สำหรับวิธีการหาผลเฉลยของปัญหากำหนดการไม่เชิงเส้นนั้นมีอยู่หลายวิธี ในปี 1951 Kuhn, H. W. และ Tucker A. W. ได้นำเสนอ Lagrange multipliers เพื่อหาผลเฉลยของปัญหากำหนดการไม่เชิงเส้นแบบที่มีเงื่อนไขบังคับแบบสมการ ในปี 1956 Frank, M. และ P. Wolfe ได้คิดค้นวิธีการใช้ทิศทางที่เป็นไปได้เพื่อใช้ในการหาผลเฉลยของปัญหากำหนดการไม่เชิงเส้นแบบที่มีเงื่อนไขบังคับเป็นอสมการเชิงเส้น ต่อมาในปี 1955 Camp, G. D. ได้คิดค้น penalty method เพื่อใช้หาผลเฉลยของปัญหากำหนดการไม่เชิงเส้นทั้งในกรณีเงื่อนไขบังคับเป็นฟังก์ชันเชิงเส้น และฟังก์ชันไม่เชิงเส้น แต่ข้อเสียของวิธีนี้คือการเลือก penalty function ให้สอดคล้องกับปัญหานั้นค่อนข้างจะยุ่งยากและซับซ้อน ในปี 1960 Kelley, J. E. ได้เสนอวิธีหาผลเฉลยปัญหากำหนดการไม่เชิงเส้นแบบมีเงื่อนไข โดยวิธีระนาบตัด (cutting-plane) จากนั้นในปี 1971 Zangwill, W. I. ได้เสนอวิธีระนาบตัด ซึ่งมีความแตกต่างจากของ Kelley, J. E. คือ วิธีระนาบตัดของ Zangwill, W. I เป็นการหาผลเฉลยของปัญหาคู่ควบที่เกิดจากฟังก์ชัน Lagrangian dual แทนปัญหากำหนดการไม่เชิงเส้นตั้งต้น

งานวิจัยนี้ได้สร้างซอฟต์แวร์สำหรับแก้ปัญหากำหนดการไม่เชิงเส้นภายใต้เงื่อนไขบังคับสมการเชิงเส้นเป็นช่วงและอสมการไม่เชิงเส้นโดยวิธีระนาบตัด ซึ่งมีรูปแบบการหาผลเฉลยที่สามารถเขียนเป็นขั้นตอนวิธีสร้างซอฟต์แวร์คอมพิวเตอร์ ขั้นตอนการหาผลเฉลยของปัญหากำหนดการไม่เชิงเส้นโดยวิธีระนาบตัดนั้นประกอบด้วยการหาผลเฉลยของปัญหากำหนดการเชิงเส้น การคำนวณค่า gradient และ Hessian ของฟังก์ชันจุดประสงค์และฟังก์ชันเงื่อนไขบังคับ และการหาผลเฉลยของระบบสมการไม่เชิงเส้น ในขั้นตอนการหาผลเฉลยของปัญหากำหนดการเชิงเส้น เราเรียกใช้คำสั่งใน library ของ GNU Linear Programming Kit (GLPK) ซึ่งเป็นคำสั่งที่ใช้เพื่อหาผลเฉลยโดยวิธี revised simplex ในขั้นตอนการหาค่า gradient และ Hessian ของฟังก์ชันจุดประสงค์และฟังก์ชันเงื่อนไขบังคับใช้หลักการ automatic differentiation โดยเรียกใช้คำสั่งใน library ของ ADOL-C และในขั้นตอนการหาผลเฉลยของปัญหากำหนดการไม่เชิงเส้นแบบมีเงื่อนไขเป็นสมการ เราใช้ Lagrange multipliers เพื่อเปลี่ยนปัญหาให้เป็นปัญหากำหนดการไม่เชิงเส้นแบบไม่มีเงื่อนไข ในขั้นตอนการหาผลเฉลยของระบบสมการไม่เชิงเส้นเราจะใช้วิธี Newton และวิธี steepest descent เพื่อนำเวลาที่ได้จากการหาผลเฉลยในแต่ละวิธีนำมาเปรียบเทียบ

วิทยานิพนธ์นี้แบ่งออกเป็น 5 บท ในบทที่ 1 จะเป็นบทนำซึ่งกล่าวถึงที่มาของการวิจัย ดำเนินงาน วัตถุประสงค์ของงานวิจัย และงานวิจัยอื่นที่เกี่ยวข้อง

บทที่ 2 กล่าวถึงความรู้พื้นฐานที่เกี่ยวกับงานวิจัย ปัญหากำหนดการเชิงเส้น ปัญหากำหนดการไม่เชิงเส้น ฟังก์ชัน Lagrangian dual ซึ่งจะอธิบายถึงความหมายทางเรขาคณิตของปัญหา Lagrangian dual และทฤษฎี strong duality รวมถึงวิธีการระนาบตัดของ Zangwill, W. I.

บทที่ 3 กล่าวถึงการสร้างซอฟต์แวร์ ขั้นตอนวิธีระนาบตัด ขั้นตอนวิธีระนาบตัดแบบขยายที่สามารถใช้หาผลเฉลยของปัญหากำหนดการไม่เชิงเส้นภายใต้เงื่อนไขบังคับอสมการไม่เชิงเส้นและสมการเชิงเส้นเป็นช่วง รวมทั้งผังงานโครงสร้างของซอฟต์แวร์ในส่วนที่เป็นโปรแกรมหลักและส่วนที่เป็นโปรแกรมย่อยที่ใช้หาผลเฉลยของปัญหาหลักและปัญหาย่อย

บทที่ 4 อธิบายถึงขั้นตอนก่อนการใช้งาน และขั้นตอนการใช้งาน แสดงตัวอย่างการเตรียมแฟ้มข้อมูลที่จะเป็นตัวกำหนดเงื่อนไขต่าง ๆ ของปัญหา และวิธีการเปลี่ยนแปลงฟังก์ชันทั้งฟังก์ชันจุดประสงค์ และฟังก์ชันเงื่อนไขบังคับ

บทที่ 5 กล่าวถึงการเปรียบเทียบผลลัพธ์ที่ได้จากซอฟต์แวร์ที่สร้างขึ้น กับผลลัพธ์ที่ได้จากโปรแกรม GAMS และสรุปผลงานวิจัย

ภาคผนวกจะแบ่งออกเป็น 2 หมวด ในหมวด ก จะอธิบายถึงวิธีการหาผลเฉลยของระบบสมการไม่เชิงเส้นโดยวิธี Newton, steepest descent และ combination of Newton and steepest descent (combine N_S) ที่นำมาใช้ในซอฟต์แวร์ และในภาคผนวก ข จะอธิบายถึงโปรแกรม Automatic Differentiation of Algorithms written in C/C++ (ADOL-C) ที่ใช้สำหรับหาค่า gradient และ Hessian ของฟังก์ชัน โปรแกรม GNU Linear Programming Kit (GLPK) ที่ใช้หาผลเฉลยของปัญหากำหนดการเชิงเส้น และโปรแกรม General Algebraic Modeling System (GAMS) ที่ใช้สำหรับหาผลเฉลยเหมาะที่สุดของปัญหากำหนดการไม่เชิงเส้น เพื่อใช้ในการเปรียบเทียบความถูกต้องและเวลาที่ใช้ในการประมวลผลกับซอฟต์แวร์ที่สร้างขึ้น ดังจะกล่าวรายละเอียดในแต่ละบทต่อไป

บทที่ 2
ความรู้พื้นฐานในงานวิจัย

ในบทนี้ จะกล่าวถึงความรู้พื้นฐานที่เกี่ยวกับงานวิจัย ปัญหากำหนดการเชิงเส้น ปัญหากำหนดการไม่เชิงเส้น ฟังก์ชัน Lagrangian dual ความหมายทางเรขาคณิตของปัญหา Lagrangian dual และทฤษฎี strong duality รวมถึงวิธีการระนาบตัดของ Kelley, J. E.

2.1 ปัญหากำหนดการเชิงเส้น

$$\begin{aligned} &\text{maximize} && z=\sum_{j=1}^{n} c_j x_j \\ &\text{subject to} && \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad : i=1,2,\ldots,m \\ &&& x_j \le 0 \end{aligned}$$

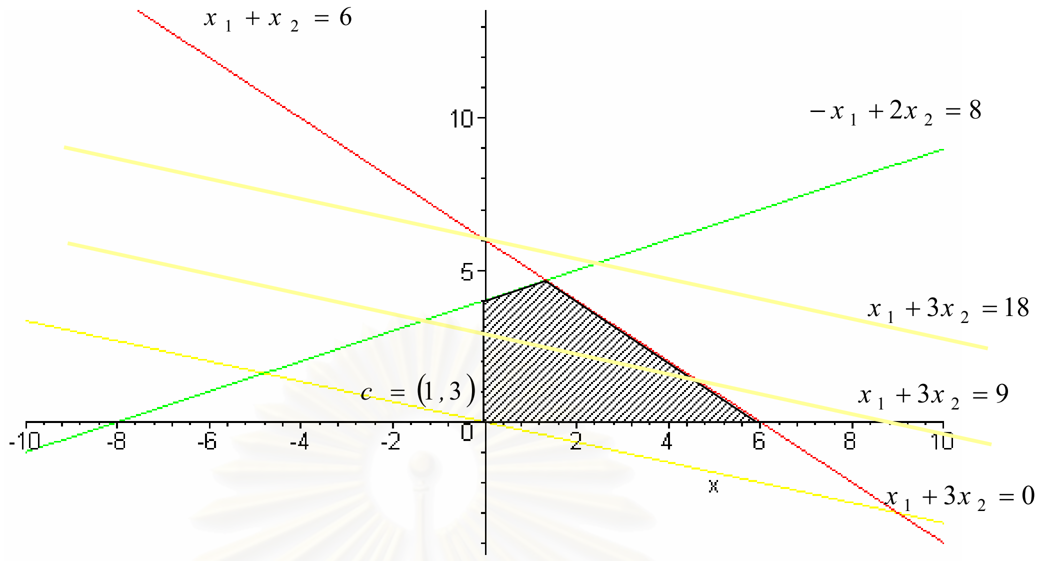
หรือ

$$\begin{aligned} &\text{minimize} && z=\sum_{j=1}^{n} c_j x_j \\ &\text{subject to} && \sum_{j=1}^{n} a_{ij} x_j \ge b_i \quad : i=1,2,\ldots,m \\ &&& x_j \le 0 \end{aligned}$$

ปัญหากำหนดการเชิงเส้นที่จะศึกษาในวิทยานิพนธ์นี้ เป็นปัญหาการหาค่ามากสุด โดยที่ z เรียกว่าค่าจุดประสงค์ $\sum_{j=1}^{n} c_j x_j$ เรียกว่าฟังก์ชันจุดประสงค์ $\sum_{j=1}^{n} a_{ij} x_j \le b_i$ เรียกว่าเงื่อนไขบังคับที่ i x_j สำหรับ $j=1,2,\ldots,n$ เรียกว่าตัวแปรตัดสินใจ a_{ij}, c_j สำหรับ $i=1,2,\ldots,m$ และ $j=1,2,\ldots,n$ เป็นสัมประสิทธิ์ค่าคงที่

<u>ตัวอย่าง 1</u>

$$\begin{aligned} &\text{minimize} && x_1 + 3x_2 \\ &\text{subject to} && x_1 + x_2 \le 6 \\ &&& -x_1 + 2x_2 \le 8 \\ &&& x_1, x_1 \ge 0 \end{aligned}$$



รูปที่ 2-1 แสดงถึงคอนทัวร์ของฟังก์ชันจุดประสงค์กับบริเวณที่เป็นไปได้

ผลเฉลยเหมาะที่สุดของปัญหากำหนดการเชิงเส้นหาได้จากการเลื่อนระนาบ $z = x_1 + 3x_2$ ไปในทิศทาง **c** ให้ไกลที่สุดโดยที่ระนาบนั้นยังคงอยู่ในบริเวณที่เป็นไปได้ ดังนั้นผลเฉลยเหมาะที่สุดจะเกิดเฉพาะที่จุดมุมของบริเวณที่เป็นไปได้เท่านั้น

2.2 ปัญหากำหนดการไม่เชิงเส้น

พิจารณาปัญหา

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{g}(\mathbf{x}) \le \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{array}$$

เมื่อ $f : \mathbf{R}^n \to \mathbf{R}$, $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^m$ และ $\mathbf{h} : \mathbf{R}^n \to \mathbf{R}^l$ โดยที่ $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ เรียกว่าตัวแปรตัดสินใจ n คือจำนวนตัวแปรตัดสินใจ ฟังก์ชัน f เรียกว่าฟังก์ชันจุดประสงค์ ฟังก์ชันอสมการ **g** และฟังก์ชันสมการ **h** เรียกว่าฟังก์ชันเงื่อนไขบังคับ

สำหรับเวกเตอร์ **x** ซึ่งสอดคล้องกับทุกเงื่อนไขบังคับเรียกว่า ผลเฉลยที่เป็นไปได้ บริเวณที่รวบรวมผลเฉลยที่เป็นไปได้เรียกว่า บริเวณที่เป็นไปได้

ปัญหานี้เรียกว่าปัญหากำหนดการไม่เชิงเส้น ถ้ามีฟังก์ชันใดฟังก์ชันหนึ่งในปัญหานี้ ไม่ว่าจะเป็นฟังก์ชันจุดประสงค์ หรือฟังก์ชันเงื่อนไขบังคับเป็นฟังก์ชันไม่เชิงเส้น แต่ถ้าเมื่อใดฟังก์ชันทุกฟังก์ชันในปัญหานี้เป็นฟังก์ชันเชิงเส้นแล้ว ปัญหานี้เรียกว่าปัญหากำหนดการเชิงเส้น

สำหรับการหาผลเฉลยของปัญหากำหนดการไม่เชิงเส้นคือการหา $\bar{\mathbf{x}}$ ที่อยู่ในบริเวณที่เป็นไปได้ที่ทำให้ $f(\bar{\mathbf{x}}) \le f(\mathbf{x})$ สำหรับทุก $\mathbf{x}$ ที่อยู่ในบริเวณที่เป็นไปได้ จุด $\bar{\mathbf{x}}$ นี้เรียกว่าผลเฉลยเหมาะที่สุด

ปัญหากำหนดการไม่เชิงเส้นที่จะศึกษาในวิทยานิพนธ์มีรูปทั่วไปดังนี้

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{g}(\mathbf{x}) \le \mathbf{0} \\ & h(\mathbf{x}) = 0 \end{array}$$

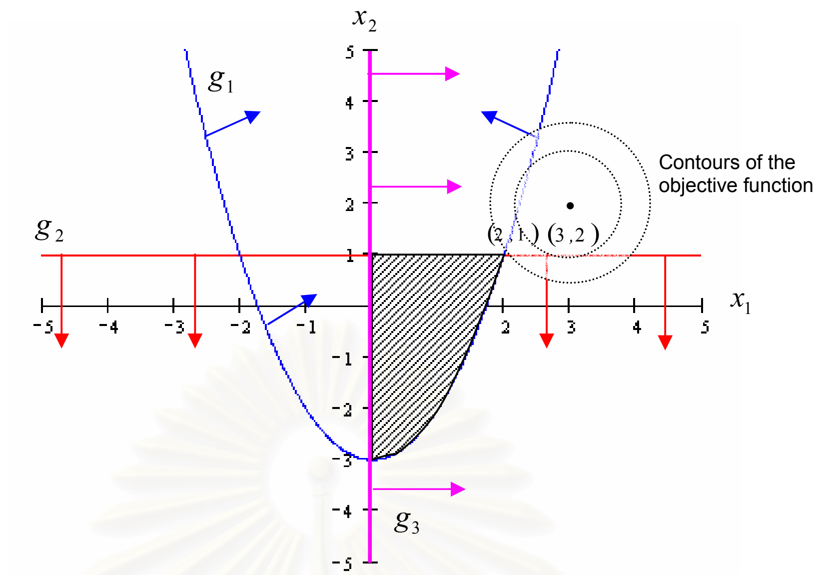
โดยที่ เป็นฟังก์ชันนูนต่อเนื่องที่สามารถหาอนุพันธ์ได้ และ $h : \mathbf{R}^n \to \mathbf{R}$ เป็นฟังก์ชันเชิงเส้น ในบางครั้งปัญหากำหนดการไม่เชิงเส้นอาจต้องการหาค่าสูงสุดของฟังก์ชันจุดประสงค์ (maximize $f(\mathbf{x})$) เราสามารถแปลงปัญหากำหนดการไม่เชิงเส้นเป็น minimize $-f(\mathbf{x})$ และสำหรับเงื่อนไขบังคับ $\mathbf{g}(\mathbf{x})$ อาจอยู่ในรูป $\mathbf{g}(\mathbf{x}) \ge \mathbf{0}$ เราสามารถแปลงเงื่อนไขบังคับให้อยู่ในรูป $-\mathbf{g}(\mathbf{x}) \le \mathbf{0}$

ตัวอย่าง 2

$$\begin{array}{ll} \text{minimize} & (x_1 - 3)^2 + (x_2 - 2)^2 \\ \text{subject to} & x_1^2 - x_2 - 3 \le 0 \\ & x_2 - 1 \le 0 \\ & -x_1 \le 0 \end{array}$$

จะเห็นได้ว่าปัญหานี้มีฟังก์ชันจุดประสงค์คือ $f(x_1,x_2) = (x_1 - 3)^2 + (x_2 - 2)^2$ และมี 3 เงื่อนไขบังคับที่เป็นอสมการ

$$g_1(x_1, x_2) = x_1^2 - x_2 - 3$$
$$g_2(x_1, x_2) = x_2 - 1$$
$$g_3(x_1, x_2) = -x_1$$



รูปที่ 2-2 แสดงถึงคอนทัวร์ของฟังก์ชันจุดประสงค์ตัดกับบริเวณที่เป็นไปได้ที่จุด (2,1) ซึ่งเป็นผลเฉลยเหมาะที่สุดของปัญหา

การหาผลเฉลยนี้คือ การหาจุดในบริเวณที่เป็นไปได้ที่ทำให้ค่า $(x_1-3)^2+(x_2-2)^2$ มีค่าน้อยที่สุด จากสมการวงกลม $(x_1-3)^2+(x_2-2)^2=c$ เมื่อ $c\geq 0$ จะเห็นได้ว่า $\sqrt{c}$ คือรัศมีของวงกลม และ (3,2) คือจุดศูนย์กลางของวงกลม เราต้องการหาวงกลมที่มีรัศมีน้อยที่สุด โดยที่เส้นรอบรูปของวงกลมนั้นยังตัดกับบริเวณที่เป็นไปได้(บริเวณแรเงาในรูป 2-2) จากรูปที่ 2-2 ค่า c ที่เล็กที่สุดคือ 2 และสัมผัสกับบริเวณที่เป็นไปได้คือ (2,1) ดังนั้นผลเฉลยเหมาะที่สุดเกิดขึ้นที่จุด (2,1) และให้ค่าจุดประสงค์เท่ากับ 2

การหาผลเฉลยโดยวิธีนี้จะไม่สะดวกเมื่อปัญหากำหนดการไม่เชิงเส้นมีตัวแปรมากกว่า 2 ตัวขึ้นไป หรือฟังก์ชันจุดประสงค์และฟังก์ชันเงื่อนไขบังคับมีความซับซ้อนมาก

2.3 Lagrangian dual

พิจารณาปัญหากำหนดการไม่เชิงเส้น

$$\begin{array}{lll} \text{primal}: & \text{minimize} & f(\mathbf{x}) \\ & \text{subject to} & \mathbf{g}(\mathbf{x}) \le \mathbf{0} \\ & & \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{array}$$

เมื่อ $f : \mathbf{R}^n \to \mathbf{R}$, $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^m$, $\mathbf{h} : \mathbf{R}^n \to \mathbf{R}^l$ และ $\mathbf{x} \in \mathbf{R}^n$ จะมีปัญหาคู่ควบที่เรียกว่าปัญหา Lagrangian dual ซึ่งอยู่ในรูปแบบ

$$\begin{array}{lll} \text{lagrangian dual}: & \text{maximize} & \theta(\mathbf{u}, \mathbf{v}) \\ & \text{subject to} & \mathbf{u} \ge \mathbf{0} \end{array}$$

เมื่อฟังก์ชัน Lagrangian dual คือ $\theta(\mathbf{u}, \mathbf{v}) = \inf\{f(\mathbf{x}) + \mathbf{u}^t \mathbf{g}(\mathbf{x}) + \mathbf{v}^t \mathbf{h}(\mathbf{x}) : \mathbf{x} \in \mathbf{R}^n\}$

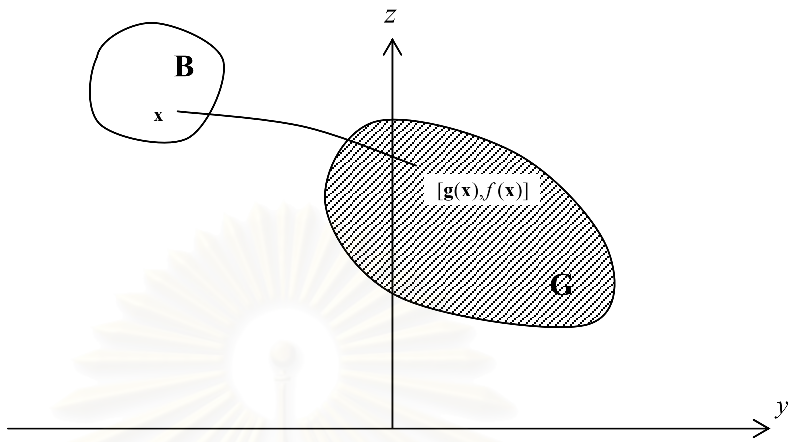
ฟังก์ชัน Lagrangian dual θ อาจจะมีค่าเข้าใกล้ $-\infty$ สำหรับบางเวกเตอร์ $(\mathbf{u}, \mathbf{v})$ จากปัญหานี้จะเห็นได้ว่าเงื่อนไขบังคับ $\mathbf{g}(\mathbf{x})$ และ $\mathbf{h}(\mathbf{x})$ ในปัญหา Primal ถูกรวมอยู่ในฟังก์ชันจุดประสงค์ในปัญหา Lagrangian dual โดยใช้ Lagrange multipliers $\mathbf{u}$ และ $\mathbf{v}$ โดยที่ $\mathbf{u}$ สัมพันธ์กับเงื่อนไขบังคับ $\mathbf{g}(\mathbf{x}) \le \mathbf{0}$ ทำให้ $\mathbf{u} \ge \mathbf{0}$ ขณะที่ $\mathbf{v}$ สัมพันธ์กับเงื่อนไขบังคับ $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ ทำให้ $\mathbf{v}$ จึงเป็นตัวแปรไม่จำกัด

2.3.1 การแปลความหมายทางเรขาคณิตของปัญหา Lagrangian dual

พิจารณากรณีที่ปัญหากำหนดการไม่เชิงเส้นมีเพียง 1 เงื่อนไขบังคับที่เป็นอสมการ นั่นคือ $g : \mathbf{R}^n \to \mathbf{R}$ และสนใจเฉพาะ $\mathbf{x} \in \mathbf{B}$ เมื่อ $\mathbf{B} \subseteq \mathbf{R}^n$ และ $\mathbf{B}$ เป็นเซตนูน ดังนั้นปัญหากำหนดการไม่เชิงเส้นจึงอยู่ในรูปแบบ

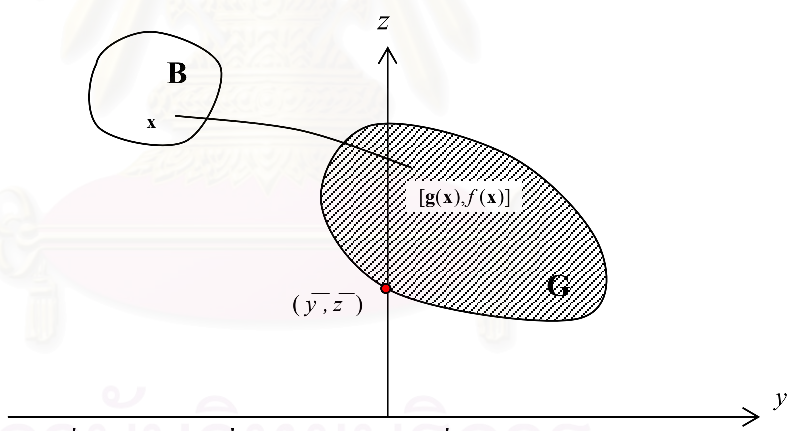
$$\begin{array}{lll} \text{P}: & \text{minimize} & f(\mathbf{x}) \\ & \text{subject to} & g(\mathbf{x}) \le 0 \\ & & \mathbf{x} \in \mathbf{B} \end{array}$$

ให้เซต G $= \{(y,z) : y = g(\mathbf{x}), z = f(\mathbf{x})$ สำหรับทุก $\mathbf{x} \in \mathbf{B}\}$ ดังแสดงในรูปที่ 2-3



รูปที่ 2-3 แสดงบริเวณที่เกิดจากการส่ง $\mathbf{x}$ ในเซต **B** ไปยังเซต **G** โดยผ่านฟังก์ชัน g, f

ดังนั้นผลเฉลยเหมาะที่สุดของปัญหา P คือการหาคู่อันดับใน G โดยที่ $y \le 0$ และทำให้ z มีค่าน้อยที่สุด จากรูปที่ 2-4 แสดงว่าคู่อันดับ $(\bar{y}, \bar{z})$ เป็นผลเฉลยของปัญหา P



รูปที่ 2-4 แสดงจุดที่เป็นผลเฉลยเหมาะที่สุดของปัญหา P

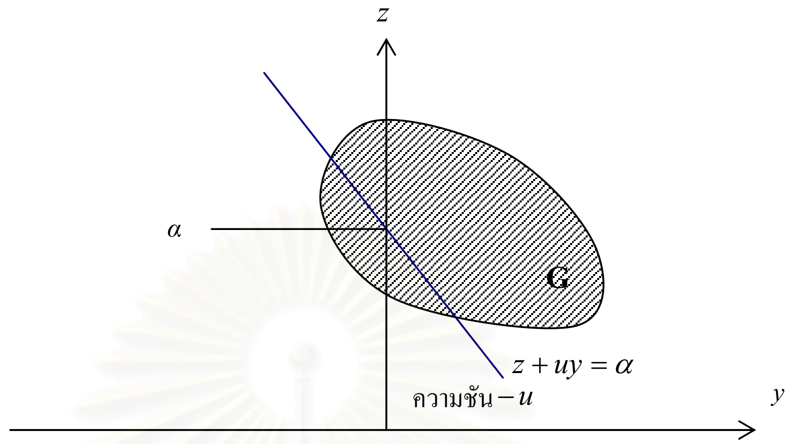
ปัญหา Lagrangian dual ที่สอดคล้องกับปัญหา P คือ

LD :	maximize	$\theta(u)$
	subject to	$u \ge 0$

เมื่อ $\theta(u) = \inf\{f(\mathbf{x}) + ug(\mathbf{x}) : \mathbf{x} \in \mathbf{B}\}$

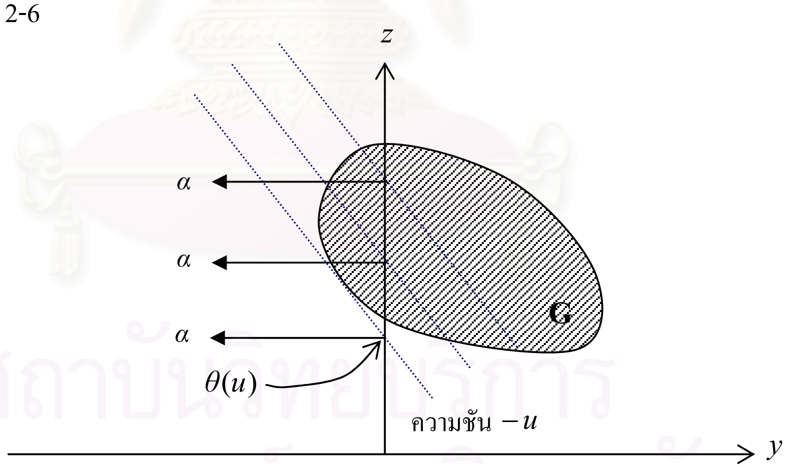
สมมุติว่าเราสามารถหาค่าที่น้อยที่สุดของ $f(\mathbf{x}) + u\,g(\mathbf{x})$ เมื่อ $\mathbf{x} \in \mathbf{B}$ ได้ ดังนั้นเราสามารถกล่าวได้ว่า $\inf\{f(\mathbf{x}) + ug(\mathbf{x}) : \mathbf{x} \in \mathbf{B}\}$ มีค่าเท่ากับ minimum$f(\mathbf{x}) + u\mathbf{g}(\mathbf{x})$ สำหรับทุก $\mathbf{x} \in \mathbf{B}$ เนื่องจาก $y = g(\mathbf{x})$ และ $z = f(\mathbf{x})$ สำหรับ $\mathbf{x} \in \mathbf{B}$ ดังนั้น $\theta(u)$ ก็คือการหาค่าน้อยที่สุด

ของ $z+uy$ บนเซต G นั่นเอง จะเห็นได้ว่า $z+uy=\alpha$ เป็นสมการเส้นตรงที่มีความชัน $-u$ และมีจุดตัดบนแกน z คือ α ดังแสดงในรูปที่ 2-5



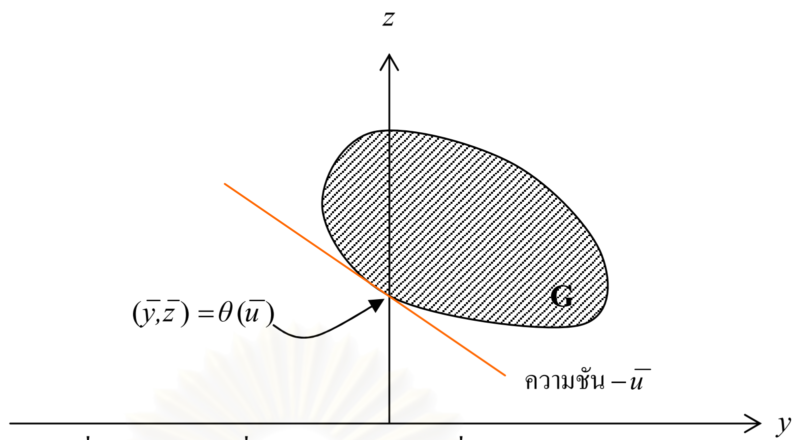
รูปที่ 2-5 แสดงสมการเส้นตรงที่ตัดแกน z ที่ α

การหาค่าน้อยที่สุดของ $z+uy$ บนเซต G ทำได้โดยเลื่อนเส้นตรง $z+uy=\alpha$ ในทิศทางที่ขนานกับเส้นตรงเดิม และทำให้จุดตัดบนแกน z มีค่าน้อยที่สุดในขณะที่เส้นตรงนั้นยังคงสัมผัสกับเซต G ซึ่งก็คือการเลื่อนเส้นตรงไปให้สัมผัสอยู่ใต้เซต G นั่นเอง ดังนั้นจุดตัดบนแกน z คือ $\theta(u)$ ดังแสดงในรูปที่ 2-6



รูปที่ 2-6 แสดงจุดตัดบนแกน z ที่เป็น $\theta(u)$

ดังนั้นปัญหา LD คือการหาความชันของ supporting hyperplane ที่ทำให้จุดตัดบนแกน z มีค่ามากที่สุด จากรูปที่ 2-7 จะเห็นได้ว่าสมการเส้นตรง $\bar{z}+\bar{u}\,\bar{y}=\bar{\alpha}$ ที่มีความชัน $-\bar{u}$ และสัมผัสเซต G ที่จุด $(\bar{y},\bar{z})$ เป็นสมการที่ให้ผลเฉลยของปัญหา LD นั่นคือมีผลเฉลยเหมาะที่สุดคือ $\bar{u}$ และมีค่าจุดประสงค์คือ $\bar{z}$ นอกจากนั้น ค่าจุดประสงค์ของปัญหา P และค่าจุดประสงค์ของปัญหา LD ยังมีค่าเท่ากันสำหรับในกรณีนี้ด้วย



รูปที่ 2-7 แสดงจุดที่เป็นผลเฉลยเหมาะที่สุดของปัญหา LD

<u>ตัวอย่างที่ 3</u>

$$\begin{aligned} \text{minimize} \quad & x_1^2 + x_2^2 \\ \text{subject to} \quad & -x_1 - x_2 + 4 \le 0 \\ & x_1, x_2 \ge 0 \end{aligned}$$

จะเห็นได้ว่าปัญหานี้มีผลเฉลยเหมาะที่สุดเกิดขึ้นที่ $(x_1, x_2) = (2,2)$ มี ค่าจุดประสงค์เท่ากับ 8

ให้ $g(\mathbf{x}) = -x_1 - x_2 + 4$ และ $\mathbf{B} = \{(x_1, x_2) : x_1, x_2 \ge 0\}$ ได้ฟังก์ชัน Lagrangian dual

$$\theta(u) = \inf\{x_1^2 + x_2^2 + u(-x_1 - x_2 + 4) : x_1, x_2 \ge 0\}$$

ดังนั้นปัญหา Lagrangian dual ที่สอดคล้องกับปัญหานี้คือ

$$\begin{aligned} \text{maximize} \quad & \theta(u) \\ \text{subject to} \quad & u \ge 0 \end{aligned}$$

จาก Kaczor, W. J. และ Nowak, M. T. [1] จะได้ว่า

$$\inf\{x_1^2 + x_2^2 + u(-x_1 - x_2 + 4) : x_1, x_2 \ge 0\} = \inf\{x_1^2 - ux_1 : x_1 \ge 0\} + \inf\{x_2^2 - ux_2 : x_2 \ge 0\} + 4u$$

ให้ $\varphi(x) = x^2 - ux$ เมื่อ $x \ge 0$ และ $u \ge 0$

จุดวิกฤตของ $\varphi(x)$ คือ $\bar{x}$ เมื่อ $\dfrac{d\varphi(\bar{x})}{dx} = 0$

$$\frac{d\varphi(\bar{x})}{dx} = 2\bar{x} - u$$

ดังนั้น $\bar{x} = \dfrac{u}{2}$ พิจารณา $\dfrac{d^2\varphi(\bar{x})}{dx^2} = 2$ แสดงว่า $\varphi(x)$ จะมีค่าต่ำสุดที่ $\bar{x}$ เนื่องจาก $\dfrac{d^2\varphi(\bar{x})}{dx^2} > 0$ ค่าขอบเขตล่างมากสุดของ φ เกิดขึ้นที่ $\bar{x} = \dfrac{u}{2}$

จึงสรุปได้ว่าค่าขอบเขตล่างมากสุดของ $\theta(u)$ เกิดขึ้นที่ $x_1 = x_2 = \dfrac{u}{2}$ เมื่อ $u \ge 0$

$$\begin{aligned}\theta(u) &= \frac{u^2}{4}+\frac{u^2}{4}+u(-\frac{u}{2}-\frac{u}{2}+4)\\ &= \frac{u^2}{2}-u^2+4u\\ &= -\frac{u^2}{2}+4u\end{aligned}$$

ดังนั้น $\theta(u) = -\frac{u^2}{2}+4u : u \geq 0$

เนื่องจาก u^2 เป็นฟังก์ชันพาราโบลาหงายซึ่งเป็นฟังก์ชันนูน ดังนั้น $-\frac{u^2}{2}$ จึงเป็นฟังก์ชันเว้า และ $4u$ เป็นฟังก์ชันเชิงเส้นจึงเป็นทั้งฟังก์ชันนูนและฟังก์ชันเว้า จาก Peressini, A. L. และ Sullivan, F. E. [2] ดังนั้น $-\frac{u^2}{2}+4u$ เป็นฟังก์ชันเว้า จึงสรุปได้ว่า $\theta(u)$ เป็นฟังก์ชันเว้าและค่าที่มากที่สุดที่เกิดบนช่วง $u \geq 0$ เกิดขึ้นที่ $\bar{u} = 4$ และ $\theta(\bar{u}) = 8$ ดังนั้น ค่าจุดประสงค์จึงมีค่าเท่ากับ 8

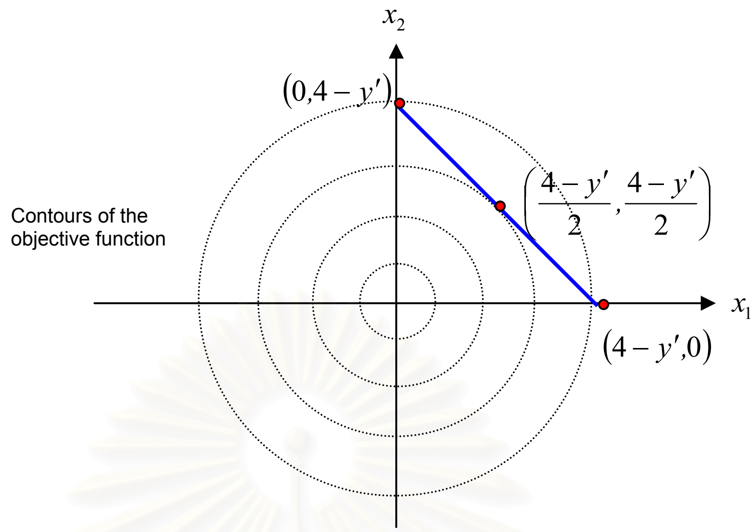
ต่อไปเราจะพิจารณาปัญหาบนระนาบ yz เมื่อ $y = \mathbf{g}(\mathbf{x})$ และ $z = f(\mathbf{x})$ เราสนใจในการหาเซต G ซึ่ง $G = \{(y,z) : y = -x_1 - x_2 + 4, z = x_1^2 + x_2^2$ สำหรับทุก $x_1, x_2 \geq 0\}$ ซึ่งกระทำได้โดยการหาขอบเขตล่างและขอบเขตบนของเซต G ซึ่งแทนด้วย $\alpha(y')$ และ $\beta(y')$ ตามลำดับ

ให้ $\alpha(y')$ และ $\beta(y')$ คือ ค่าจุดประสงค์ที่เหมาะที่สุดของปัญหา P1 และ P2 ตามลำดับ

P1 : minimize $x_1^2 + x_2^2$
subject to $-x_1 - x_2 + 4 = y'$
$x_1, x_2 \geq 0$

P2 : maximize $x_1^2 + x_2^2$
subject to $-x_1 - x_2 + 4 = y'$
$x_1, x_2 \geq 0$

จากเงื่อนไขบังคับของทั้งสองปัญหา $-x_1 - x_2 + 4 = y'$ จะได้ $x_1 + x_2 = 4 - y'$ และ $x_1, x_2 \geq 0$ ดังนั้น $y' \leq 4$ และเมื่อค่า y' ลดลง $x_1 + x_2$ จะมีค่าเพิ่มมากขึ้น



รูปที่ 2-8 ผลเฉลยทางเรขาคณิตของปัญหากำหนดการไม่เชิงเส้น P1 และ P2

ให้ฟังก์ชันจุดประสงค์ $\eta(x_1,x_2)=x_1^2+x_2^2$ จาก $x_1+x_2=4-y'$ จะได้ $x_2=4-y'-x_1$ แทน x_2 ลงใน $\eta(x_1,x_2)$ ดังนั้นตัวแปรในฟังก์ชันจุดประสงค์จะถูกลดลงเหลือเพียงตัวแปรเดียว จะได้

$$\begin{aligned}\eta(x_1) &= x_1^2+(4-y'-x_1)^2\\ &= x_1^2+(4-y')^2-2x_1(4-y')+x_1^2\\ &= 2x_1^2-2x_1(4-y')+(4-y')^2\end{aligned}$$

ดังนั้น $\eta(x_1)=2x_1^2-2x_1(4-y')+(4-y')^2$

เนื่องจาก y' เป็นค่าคงที่ใน $\eta(x_1)$ ดังนั้น $-2x_1(4-y')+(4-y')^2$ เป็นฟังก์ชันเชิงเส้นจึงเป็นทั้งฟังก์ชันนูนและฟังก์ชันเว้า และ $2x_1^2$ เป็นฟังก์ชันพาราโบลาหงายซึ่งเป็นฟังก์ชันนูน จึงสรุปได้ว่า $\eta(x_1)$ เป็นฟังก์ชันนูน

จุดวิกฤตของ $\eta(x_1)$ จะเกิดขึ้นที่ $\bar{x}_1$ เมื่อ $\frac{d\eta(\bar{x}_1)}{dx_1}=0$

$$\frac{d\eta(\bar{x}_1)}{dx_1}=4\bar{x}_1-2(4-y')$$

$\bar{x}_1=\frac{2(4-y')}{4}=\frac{(4-y')}{2}$ จากนั้นตรวจสอบ $\frac{d^2\eta(\bar{x}_1)}{dx_1^2}$ ถ้า $\frac{d^2\eta(\bar{x}_1)}{dx_1^2}>0$ แล้ว $\eta(x_1)$ จะมีค่าต่ำสุดที่ $\bar{x}_1$

$$\frac{d^2\eta(\bar{x}_1)}{dx_1^2} = 4$$

ดังนั้นที่ $\bar{x}_1 = \dfrac{(4 - y')}{2}$ จะทำให้ $\eta(x_1)$ มีค่าต่ำสุดวงกว้าง

$$\bar{x}_2 = 4 - y' - \frac{(4 - y')}{2}$$

$$= \frac{(4 - y')}{2}$$

จึงสรุปได้ว่าจุด $\left(\dfrac{(4 - y')}{2}, \dfrac{(4 - y')}{2}\right)$ เป็นจุดที่ทำให้ $\eta(x_1, x_2)$ มีค่าต่ำสุด ดังนั้นค่าจุดประสงค์จะมีค่าเท่ากับ

$$\left(\frac{4 - y'}{2}\right)^2 + \left(\frac{4 - y'}{2}\right)^2 = \frac{(4 - y')^2}{2}$$

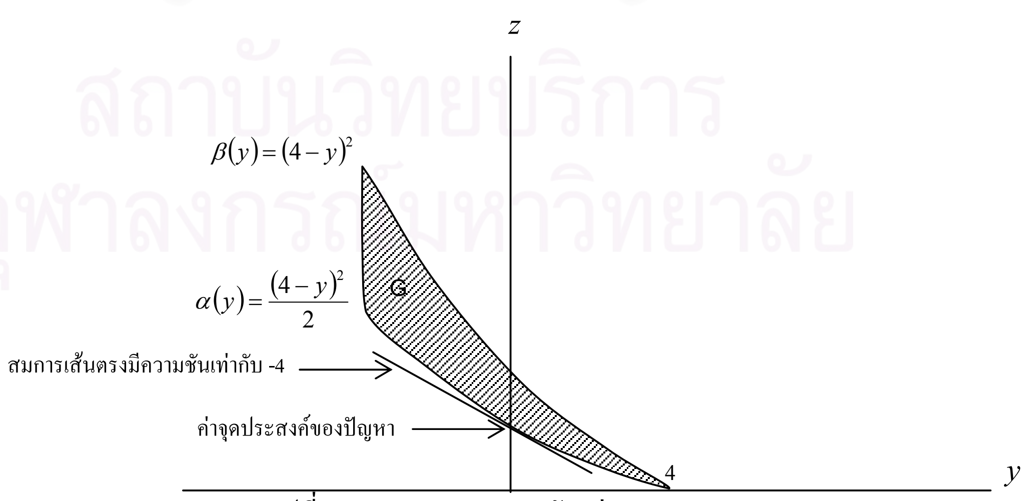
กล่าวคือ $\alpha(y') = \dfrac{(4 - y')^2}{2}$

เนื่องจากจุดวิกฤตของ $\eta(x_1, x_2)$ มี 3 จุดคือ จุดขอบ $(4 - y', 0), (0, 4 - y')$ และจุด $\left(\dfrac{(4 - y')}{2}, \dfrac{(4 - y')}{2}\right)$ จะพิจารณาอีก 2 จุดที่เหลือ

ที่จุด $(4 - y', 0)$, $\eta(x_1, x_2) = (4 - y')^2 + 0 = (4 - y')^2$

ที่จุด $(0, 4 - y')$, $\eta(x_1, x_2) = 0 + (4 - y')^2 = (4 - y')^2$

ดังนั้น 2 จุดนี้คือจุดที่ทำให้ $\eta(x_1, x_2)$ มีค่ามากที่สุด กล่าวคือ $\beta(y') = (4 - y')^2$ เซต G จึงมีลักษณะดังรูป 2-9



รูปที่ 2-9 แสดงเซต G ของตัวอย่าง 2

จากรูปที่ 2-9 จะเห็นได้ว่า สมการเส้นตรงที่มีความชันเท่ากับ –4 สัมผัสกับเซต G ที่จุด (0,8) เป็นสมการเส้นตรงที่ให้ผลเฉลยของปัญหา Lagrangian dual นั่นคือมีผลเฉลยเหมาะที่สุดเท่ากับ 4 และมีค่าจุดประสงค์เท่ากับ 8

2.3.2 Strong Duality Theorem

ให้ $f:\mathbf{R}^n \to \mathbf{R}$ และ $\mathbf{g}:\mathbf{R}^n \to \mathbf{R}^m$ เป็นฟังก์ชันนูน และ $\mathbf{h}:\mathbf{R}^n \to \mathbf{R}^l$ เป็นฟังก์ชันเชิงเส้น ถ้ามี $\hat{\mathbf{x}}$ ซึ่ง $\mathbf{g}(\hat{\mathbf{x}}) \le \mathbf{0}, \mathbf{h}(\hat{\mathbf{x}}) = \mathbf{0}$ แล้ว

$$\inf\{f(\mathbf{x}) : \mathbf{g}(\mathbf{x}) \le \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}\} = \sup\{\theta(\mathbf{u},\mathbf{v}) : \mathbf{u} \ge \mathbf{0}\}$$

ถ้าเรากำหนดเงื่อนไขของปัญหา เพื่อกำจัดผลต่างระหว่างค่าจุดประสงค์ของปัญหา primal และ Lagrangian dual แล้ว เราสามารถหาผลเฉลยของปัญหา primal โดยหาผลเฉลยของปัญหา Lagrangian dual แทน สำหรับการหาผลเฉลยของปัญหา Lagrangian dual ในวิทยานิพนธ์นี้เราจะใช้วิธี cutting-plane โดยมีขั้นตอนวิธีดังนี้

2.3.3 วิธี Cutting-Plane

พิจารณาปัญหากำหนดการไม่เชิงเส้น

$$
\begin{aligned}
(\mathbf{I}) \qquad & \text{minimize } f(\mathbf{x}) \\
& \text{subject to } \quad \mathbf{g}(\mathbf{x}) \le \mathbf{0} \\
& \qquad\qquad\quad\; \mathbf{h}(\mathbf{x}) = \mathbf{0}
\end{aligned}
$$

เมื่อ $f:\mathbf{R}^n \to \mathbf{R}, g:\mathbf{R}^n \to \mathbf{R}^m, h:\mathbf{R}^n \to \mathbf{R}^l$ เมื่อ $\mathbf{x} \in \mathbf{R}^n$ สำหรับการหาผลเฉลยของปัญหา $(\mathbf{I})$ โดยวิธี cutting-plane จะไม่ได้หาผลเฉลยของปัญหาโดยตรง แต่จะหาผลเฉลยของปัญหาคู่ควบที่เรียกว่าปัญหา Lagrangian dual $(\mathbf{II})$

$$
\begin{aligned}
(\mathbf{II}) \qquad & \text{maximize } \quad \theta(\mathbf{u},\mathbf{v}) \\
& \text{subject to } \quad \mathbf{u} \ge \mathbf{0}
\end{aligned}
$$

โดยที่ $\theta(\mathbf{u},\mathbf{v}) = \inf\{f(\mathbf{x}) + \mathbf{u}^t\mathbf{g}(\mathbf{x}) + \mathbf{v}^t\mathbf{h}(\mathbf{x}) : \mathbf{x} \in \mathbf{R}^n\}$

เมื่อเราสมมุติให้ $z = \theta(\mathbf{u},\mathbf{v})$ เพราะฉะนั้น $z \le f(\mathbf{x}) + \mathbf{u}^t\mathbf{g}(\mathbf{x}) + \mathbf{v}^t\mathbf{h}(\mathbf{x})$ สำหรับทุก $\mathbf{x} \in \mathbf{R}^n$ ดังนั้นปัญหา maximize $\theta(\mathbf{u},\mathbf{v})$ โดยที่ $\mathbf{u} \ge \mathbf{0}$ จึงสมมูลกับปัญหา $(\mathbf{III})$

$$
\begin{aligned}
(\mathbf{III}) \qquad & \text{maximize } \; z \\
& \text{subject to } \quad z \le f(\mathbf{x}) + \mathbf{u}^t\mathbf{g}(\mathbf{x}) + \mathbf{v}^t\mathbf{h}(\mathbf{x}) \quad \text{ทุก } \mathbf{x} \in \mathbf{R}^n \\
& \qquad\qquad \mathbf{u} \ge \mathbf{0}
\end{aligned}
$$

จะเห็นได้ว่าปัญหา (**III**) เป็นปัญหากำหนดการเชิงเส้นที่มีจำนวนเงื่อนไขบังคับไม่จำกัด ซึ่งยากต่อการหาผลเฉลย ดังนั้นเราจะพิจารณาหาผลเฉลยของปัญหา (**IV**) แทนปัญหา (**III**)

(IV) maximize z

subject to $z \le f(\mathbf{x}^{(j)}) + \mathbf{u}^t\mathbf{g}(\mathbf{x}^{(j)}) + \mathbf{v}^t\mathbf{h}(\mathbf{x}^{(j)})$; $j = 0,1,\ldots,k-1$

$\mathbf{u} \ge \mathbf{0}$

เมื่อ $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k-1)} \in \mathbf{R}^n$ จะเห็นว่าปัญหา (**IV**) เป็นปัญหากำหนดการเชิงเส้นที่มีจำนวนเงื่อนไขบังคับจำกัด สมมุติว่าได้ผลเฉลยเป็น $(z^{(k)}, \mathbf{u}^{(k)}, \mathbf{v}^{(k)})$ แต่เรายังไม่สามารถยืนยันได้ว่า $(z^{(k)}, \mathbf{u}^{(k)}, \mathbf{v}^{(k)})$ จะเป็นผลเฉลยของปัญหา (**III**) ด้วย เนื่องจากปัญหา (**IV**) นี้ เราพิจารณาเพียงแค่ $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k-1)}$ เท่านั้นอาจจะมี $\mathbf{x}' \in \mathbf{R}^n$ ที่ไม่ใช่ $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k-1)}$ ที่ทำให้ z ขัดแย้งกับเงื่อนไขบังคับในปัญหา (**III**) นั่นคือ $z > f(\mathbf{x}') + \mathbf{u}^t\mathbf{g}(\mathbf{x}') + \mathbf{v}^t\mathbf{h}(\mathbf{x}')$ ดังนั้นเราจึงต้องทดสอบว่าผลเฉลยจากปัญหา (**IV**) สอดคล้องกับทุกเงื่อนไขบังคับของปัญหา (**III**) หรือไม่ โดยพิจารณาหาผลเฉลยของปัญหา (**V**)

(V) minimize $f(\mathbf{x}) + \mathbf{u}^{(k)t}\mathbf{g}(\mathbf{x}) + \mathbf{v}^{(k)t}\mathbf{h}(\mathbf{x})$

subject to $\mathbf{x} \in \mathbf{R}^n$

สมมุติได้ $\mathbf{x}^{(k)}$ เป็นผลเฉลยของปัญหา (**V**) นั่นหมายความว่า ที่ $(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^{(k)}, \mathbf{v}^{(k)})$ ค่า $\mathbf{x}$ ที่ทำให้ $f(\mathbf{x}) + \mathbf{u}^{(k)t}\mathbf{g}(\mathbf{x}) + \mathbf{v}^{(k)t}\mathbf{h}(\mathbf{x})$ มีค่าน้อยที่สุดคือ $\mathbf{x}^{(k)}$ ดังนั้น $\theta^{(k)} = f(\mathbf{x}^{(k)}) + \mathbf{u}^{(k)t}\mathbf{g}(\mathbf{x}^{(k)}) + \mathbf{v}^{(k)t}\mathbf{h}(\mathbf{x}^{(k)})$ ถ้า $z^{(k)} \le \theta^{(k)}$ แล้วแสดงว่า $z^{(k)}$ สอดคล้องกับทุกเงื่อนไขบังคับของปัญหา (**III**) ทำให้ $(\mathbf{u}^{(k)}, \mathbf{v}^{(k)})$ เป็นผลเฉลยของปัญหา (**II**) ในทางปฏิบัติเราสามารถหาค่า $z^{(k)}$ ที่มีค่าใกล้เคียง $\theta^{(k)}$ มากพอที่จะยอมรับได้นั่นคือ $\left|z^{(k)} - \theta^{(k)}\right| \le \varepsilon$ โดยที่ ε เป็นจำนวนจริงที่มีน้อยมาก แต่ถ้า $\left|z^{(k)} - \theta^{(k)}\right| > \varepsilon$ แสดงว่า $(z^{(k)}, \mathbf{u}^{(k)}, \mathbf{v}^{(k)})$ ผลเฉลยนี้ไม่สอดคล้องกับเงื่อนไขของปัญหา (**III**) ดังนั้นเราจึงต้องเพิ่มเงื่อนไขบังคับ $z \le f(\mathbf{x}^{(k)}) + \mathbf{u}^t\mathbf{g}(\mathbf{x}^{(k)}) + \mathbf{v}^t\mathbf{h}(\mathbf{x}^{(k)})$ ในปัญหา (**IV**) แล้วกลับไปหาผลเฉลยของปัญหา (**IV**) ใหม่ ทำเช่นนี้เรื่อยไปจนกระทั่งได้ผลเฉลยของปัญหา (**II**) หรือจำนวนครั้งของการทำซ้ำมากกว่าที่กำหนดไว้

บทที่ 3
การสร้างซอฟต์แวร์

ในบทนี้เราจะกล่าวถึงรายละเอียดของขั้นตอนวิธี cutting-plane และผังงานของซอฟต์แวร์ที่ใช้หาผลเฉลยที่เหมาะสมที่สุดของปัญหากำหนดการไม่เชิงเส้นภายใต้เงื่อนไขบังคับอสมการไม่เชิงเส้นและสมการเชิงเส้นเป็นช่วง การสร้างซอฟต์แวร์ในวิทยานิพนธ์นี้ได้ใช้ภาษาซี เนื่องจากเราสามารถเรียกใช้ซอฟต์แวร์ที่ใช้หาผลเฉลยของปัญหากำหนดการเชิงเส้น และซอฟต์แวร์ที่ใช้หาอนุพันธ์ของฟังก์ชันที่เขียนโดยใช้ภาษาซีอยู่แล้ว

3.1 ขั้นตอนวิธีระนาบตัด (Cutting-Plane Algorithm)

$$
\begin{array}{ll}
\text{minimize} & f(\mathbf{x}) \\
\text{subject to} & \mathbf{g}(\mathbf{x}) \le \mathbf{0} \\
 & \mathbf{h}(\mathbf{x}) = 0
\end{array}
$$

ให้ $\mathbf{x} \in \mathbf{R}^n$, $f : \mathbf{R}^n \to \mathbf{R}$ และ $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^m$ เป็นฟังก์ชันนูนต่อเนื่องที่สามารถหาอนุพันธ์ได้ และ $\mathbf{h} : \mathbf{R}^n \to \mathbf{R}^l$ เป็นฟังก์ชันเชิงเส้น ปัญหา Lagrangian dual ที่สอดคล้องกับปัญหานี้คือ

$$
\begin{array}{ll}
\text{maximize} & \theta(\mathbf{u}, \mathbf{v}) \\
\text{subject to} & \mathbf{u} \ge \mathbf{0}
\end{array}
$$

โดยที่ $\theta(\mathbf{u}, \mathbf{v}) = \inf\{ f(\mathbf{x}) + \mathbf{u}^t \mathbf{g}(\mathbf{x}) + \mathbf{v}^t \mathbf{h}(\mathbf{x}) : \mathbf{x} \in \mathbf{R}^n \}$

<u>ขั้นตอนเริ่มต้น</u>

กำหนด $\mathbf{x}^{(0)}$ ซึ่ง $\mathbf{g}(\mathbf{x}^{(0)}) \le \mathbf{0}$ และ $\mathbf{h}(\mathbf{x}^{(0)}) = 0$ ให้ $k = 1$ แล้วเข้าสู่ขั้นตอนหลัก

โดยที่

$$
\mathbf{g}(\mathbf{x}^{(0)}) = \begin{bmatrix} \mathrm{gg}_1(\mathbf{x}^{(0)}) \\ \mathrm{g}_2(\mathbf{x}^{(0)}) \\ \vdots \\ \mathrm{g}_m(\mathbf{x}^{(0)}) \end{bmatrix}
$$

ขั้นตอนหลัก

ขั้นที่ 1 : หาผลเฉลยที่เหมาะที่สุดของปัญหาหลัก

$$\begin{aligned} &\text{maximize} && z \\ &\text{subject to} && z \le f(\mathbf{x}^{(j)}) + \mathbf{u}^t \mathbf{g}(\mathbf{x}^{(j)}) + \mathbf{v}^t \mathbf{h}(\mathbf{x}^{(j)}) \quad ; j = 0,1,\dots,k-1 \\ &&& \mathbf{u} \ge \mathbf{0} \end{aligned}$$

ให้ $(z^{(k)}, \mathbf{u}^{(k)}, \mathbf{v}^{(k)})$ เป็นผลเฉลยที่เหมาะที่สุดของปัญหาหลัก

ขั้นที่ 2 : หาผลเฉลยที่เหมาะที่สุดของปัญหาย่อย

$$\begin{aligned} &\text{minimize} \ f(\mathbf{x}) + \mathbf{u}^{(k)^t} \mathbf{g}(\mathbf{x}) + \mathbf{v}^{(k)^t} \mathbf{h}(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{x} \in \mathbf{R}^n \end{aligned}$$

ให้ $\mathbf{x}^{(k)}$ เป็นผลเฉลยที่เหมาะที่สุดของปัญหาย่อย

ขั้นที่ 3 ให้ $\theta^{(k)} = f(\mathbf{x}^{(k)}) + \mathbf{u}^{(k)^t} \mathbf{g}(\mathbf{x}^{(k)}) + \mathbf{v}^{(k)^t} \mathbf{h}(\mathbf{x}^{(k)})$

ขั้นที่ 4 ถ้า $\left|z^{(k)} - \theta^{(k)}\right| < \varepsilon$ แล้วหยุด $(\theta^{(k)}, \mathbf{u}^{(k)}, \mathbf{v}^{(k)})$ เป็นค่าประมาณของผลเฉลยที่เหมาะที่สุดของปัญหา lagrangian dual เมื่อ ε เป็นจำนวนจริงบวกที่มีขนาดเล็ก ถ้าไม่เช่นนั้นให้เพิ่มเงื่อนไขบังคับ $z \le f(\mathbf{x}^{(k)}) + \mathbf{u}^t \mathbf{g}(\mathbf{x}^{(k)}) + \mathbf{v}^t \mathbf{h}(\mathbf{x}^{(k)})$ ในปัญหาหลัก เพิ่มค่า $k = k + 1$ แล้วกลับไปขั้นตอนหลักอีกครั้ง

3.2 ขั้นตอนวิธีระนาบตัดแบบขยายออก (Extended Cutting Plane Algorithm)

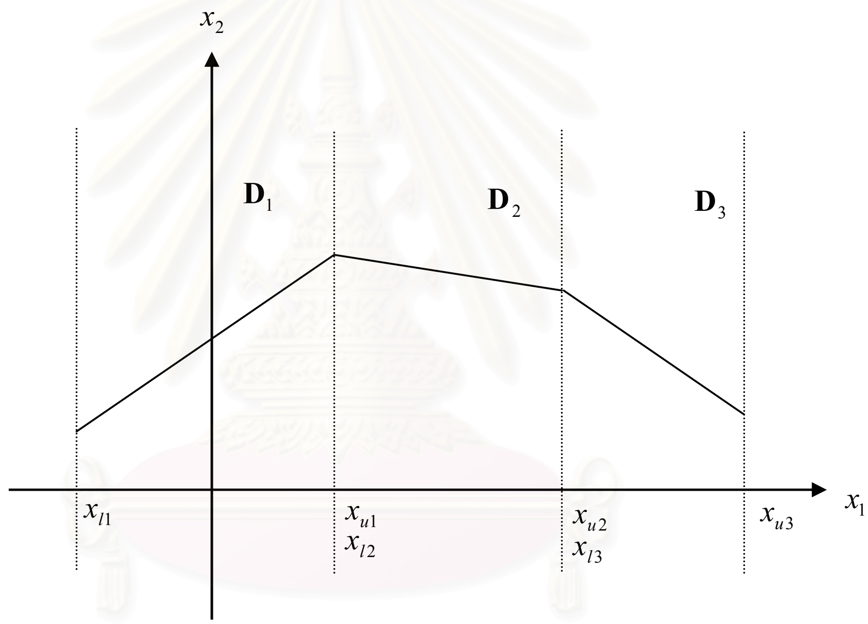
เราได้ดัดแปลงและขยายวิธีการระนาบตัดให้หาผลเฉลยได้ง่ายขึ้น ทั้งยังสามารถใช้ในกรณีที่ปัญหากำหนดการไม่เชิงเส้นมีเงื่อนไขบังคับเป็นสมการเชิงเส้นเป็นช่วงได้ ปัญหากำหนดการไม่เชิงเส้นที่จะศึกษาในวิทยานิพนธ์มีรูปแบบดังนี้

$$\begin{aligned} \text{P}: \quad &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && \mathbf{g}(\mathbf{x}) \le \mathbf{0} \\ &&& h(\mathbf{x}) = 0 \end{aligned}$$

เมื่อ $\mathbf{x} \in \mathbf{D}$ และ $f : \mathbf{D} \to \mathbf{R}$, $\mathbf{g} : \mathbf{D} \to \mathbf{R}^m$ เป็นฟังก์ชันนูนต่อเนื่องที่สามารถหาอนุพันธ์ได้ และ $h : \mathbf{D} \to \mathbf{R}$ โดยที่ $\mathbf{D} \subseteq \mathbf{R}^n$ ในกรณีที่ h เป็นฟังก์ชันเชิงเส้นจะมีเงื่อนไขดังนี้

- h เป็นฟังก์ชันต่อเนื่อง
- ให้ $a \le x_1 \le b$ และ $t_0, t_1, \ldots, t_l$ โดยที่ $a = t_0 < t_1 < \ldots < t_l = b$ แล้ว $\mathbf{D}_i = [t_{i-1}, t_i] \times \mathbf{R}^{n-1}$ สำหรับ $i = 1, 2, \ldots, l$ โดยที่ $\bigcup_{i=1}^{l} \mathbf{D}_i = \mathbf{D}$ และ $\mathrm{int}(\mathbf{D}_i) \cap \mathrm{int}(\mathbf{D}_j) = \varphi$ เมื่อ $i \ne j$ และ $\mathrm{int}(\mathbf{D}_i)$ หมายถึง $\mathbf{x}$ ทุกตัวที่อยู่ใน $\mathbf{D}_i$
- $h(\mathbf{x}) = \mathbf{p}_i^t \mathbf{x} + c_i$ ถ้า $\mathbf{x} \in \mathbf{D}_i$ เมื่อ $\mathbf{p}_i \in \mathbf{R}^n, c_i \in \mathbf{R}$

ผลเฉลยที่เหมาะที่สุดที่ได้จากแต่ละช่วงจะเป็นค่าต่ำสุดเฉพาะที่ ถ้านำผลเฉลยที่เหมาะที่สุดที่ได้จากแต่ละช่วงมาเปรียบเทียบกัน ผลเฉลยที่มีค่าน้อยที่สุดจะเป็นผลเฉลยที่เหมาะที่สุดสำหรับปัญหากำหนดการไม่เชิงเส้น กล่าวคือ ผลเฉลยนี้เป็นค่าต่ำสุดวงกว้างนั่นเอง



รูปที่3.1 แสดงถึงตัวอย่างโดเมนของ $\mathbf{x}$

ขั้นตอนวิธีสำหรับวิธีระนาบตัดแบบขยายออก

ขั้นที่ 1: แบ่งปัญหากำหนดการไม่เชิงเส้นออกเป็นช่วงตามเงื่อนไขบังคับสมการเชิงเส้น

ขั้นที่ 2: หาผลเฉลยของปัญหากำหนดการไม่เชิงในแต่ละช่วง

ขั้นที่ 3: นำผลเฉลยที่ได้จากแต่ละช่วงมาเปรียบเทียบ

ปัญหากำหนดการไม่เชิงเส้นในช่วงที่ $\mathbf{x} \in \mathbf{D}_i$ มีรูปแบบดังนี้

ปัญหา (I)

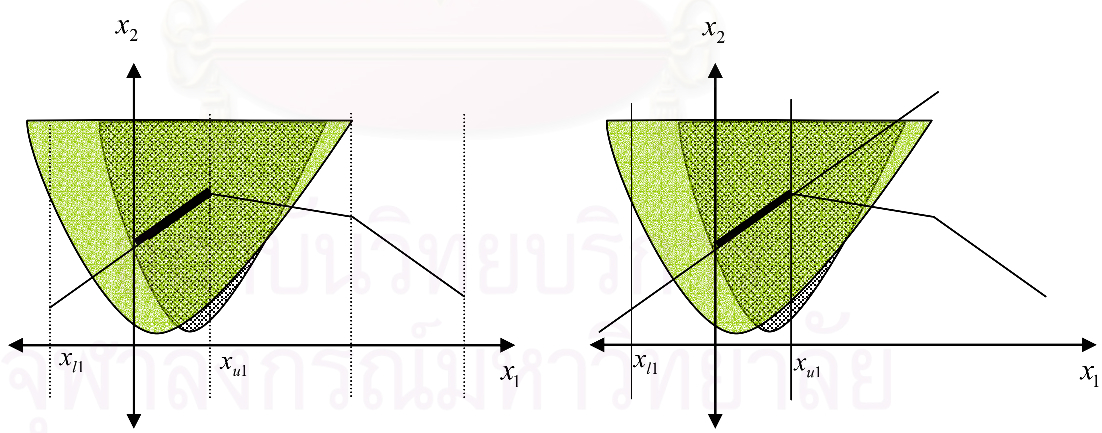
$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ & h(\mathbf{x}) = 0 \\ & \mathbf{x} \in \mathbf{D}_i \end{aligned}$$

โดยที่ $\mathbf{D}_i = \{x_1 | x_{li} \leq x_1 \leq x_{ui}\} \times \mathbf{R}^{n-1}$ และ $f : \mathbf{D}_i \to \mathbf{R}$, $\mathbf{g} : \mathbf{D}_i \to \mathbf{R}^m$ เป็นฟังก์ชันนูนต่อเนื่องที่สามารถหาอนุพันธ์ได้และ $\mathbf{h} : \mathbf{D}_i \to \mathbf{R}$ โดยที่ $h(\mathbf{x}) = \mathbf{p}_i^t \mathbf{x} + c_i$ เมื่อ $\mathbf{p}_i \in \mathbf{R}^n, c_i \in \mathbf{R}$ ดังนั้นปัญหากำหนดการไม่เชิงเส้นจึงอยู่ในรูป

ปัญหา (II)

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ & -x_1 + x_{li} \leq 0 \\ & x_1 - x_{ui} \leq 0 \\ & h(\mathbf{x}) = \mathbf{p}_i^t \mathbf{x} + c_i = 0 \end{aligned}$$

เราจะเพิ่มฟังก์ชัน $-x_1 + x_{li}$ และ $x_1 - x_{ui}$ ลงใน $\mathbf{g}$ ดังนั้น $\mathbf{g} : \mathbf{D}_i \to \mathbf{R}^{m+2}$ โดเมนของ $\mathbf{x}$ สามารถเปลี่ยนจาก $\mathbf{D}_i$ เป็น $\mathbf{R}^n$ โดยไม่ทำให้บริเวณที่เป็นไปได้เปลี่ยนไป เนื่องจากเราเพิ่มเงื่อนไข $-x_1 + x_{li} \leq 0$ และ $x_1 - x_{ui} \leq 0$ ลงในปัญหากำหนดการไม่เชิงเส้นดังรูป 3.2



รูปที่ 3.2 แสดงถึงบริเวณที่เป็นไปได้ของปัญหา (I) และปัญหา (II)

จาก Bazaraa, Sherali and Shetty [3] ให้ $\mathbf{D}' = \{\mathbf{x} \in \mathbf{R}^n : h(\mathbf{x}) = \mathbf{p}_i^t \mathbf{x} + c_i = 0\}$ ดังนั้นปัญหากำหนดการไม่เชิงเส้นจึงอยู่ในรูป

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && \mathbf{g}(\mathbf{x}) \le \mathbf{0} \\ &&& \mathbf{x} \in \mathbf{D}' \end{aligned}$$

ดังนั้นปัญหา lagrangian dual ที่สอดคล้องกับปัญหานี้คือ

$$\begin{aligned} &\text{maximize} && \theta(\mathbf{u}) \\ &\text{subject to} && \mathbf{u} \ge \mathbf{0} \end{aligned}$$

โดยที่ $\theta(\mathbf{u}) = \inf\{ f(\mathbf{x}) + \mathbf{u}^t \mathbf{g}(\mathbf{x}) : \mathbf{x} \in \mathbf{D}' \}$

<u>ขั้นตอนเริ่มต้น</u>

กำหนด $\mathbf{x}^{(0)} \in \mathbf{D}'$, $\mathbf{g}(\mathbf{x}^{(0)}) \le \mathbf{0}$ ให้ $k = 1$ และเข้าสู่ขั้นตอนหลัก

<u>ขั้นตอนหลัก</u>

ขั้นที่ 1: หาผลเฉลยที่เหมาะที่สุดของปัญหาหลัก

$$\begin{aligned} &\text{maximize} && z \\ &\text{subject to} && z \le f(\mathbf{x}^{(j)}) + \mathbf{u}^t \mathbf{g}(\mathbf{x}^{(j)}) \quad ; j = 0, 1, \ldots, k-1 \\ &&& \mathbf{u} \ge \mathbf{0} \end{aligned}$$

ให้ $(z^{(k)}, \mathbf{u}^{(k)})$ เป็นผลเฉลยที่เหมาะที่สุดของปัญหาหลัก เราสังเกตได้ว่าเมื่อ $k = 1$ แล้วผลเฉลยที่เหมาะที่สุดจะได้ $z^{(1)} = f(\mathbf{x}^{(1)}), \mathbf{u}^{(1)} = \mathbf{0}$ เพราะว่า $\mathbf{g}(\mathbf{x}^{(0)}) \le \mathbf{0}$ ดังนั้น การที่จะหาค่า z ที่มากที่สุดต้องให้ $\mathbf{u} = \mathbf{0}$ เพราะถ้า $\mathbf{u}$ เพิ่มขึ้น ค่า z จะลดลง หลังจากนั้นเมื่อ k เพิ่มขึ้นเรื่อย ๆ บริเวณที่เป็นไปได้ของปัญหาจะมีพื้นที่ลดลงอาจจะเป็นได้ทั้งกรณีที่พื้นที่มีขอบเขต(bounded) หรือพื้นที่ไม่มีขอบเขต(unbounded) ซึ่งจะขึ้นอยู่กับเงื่อนไขบังคับที่เพิ่มเข้ามาในแต่ละรอบที่เพิ่มค่า k แต่ทุกครั้งเราสามารถหาผลเฉลยที่เหมาะที่สุดได้เนื่องจากค่าจุดประสงค์ของปัญหาขึ้นอยู่กับตัวแปร z เพียงตัวเดียวเท่านั้น นั่นคือเราต้องหาจุดขอบของบริเวณที่เป็นไปได้ที่ทำให้ค่า z มีค่ามากที่สุด

ขั้นที่ 2: หาผลเฉลยที่เหมาะที่สุดของปัญหาย่อย

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) + \mathbf{u}^{(k)t} \mathbf{g}(\mathbf{x}) \\ &\text{subject to} && \mathbf{x} \in \mathbf{D}' \end{aligned}$$

สามารถเขียนได้เป็น

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) + \mathbf{u}^{(k)t} \mathbf{g}(\mathbf{x}) \\ &\text{subject to} && h(\mathbf{x}) = 0 \end{aligned}$$

จะเห็นได้ว่าปัญหานี้เป็นปัญหากำหนดการไม่เชิงเส้น ที่มีเงื่อนไขบังคับเป็นสมการเชิงเส้น ดังนั้นเราสามารถหาอนุพันธ์ย่อยอันดับที่หนึ่งของเงื่อนไขบังคับได้ จาก Frederick Hiller, Gerald Lieberman [4] เราสามารถลดรูปปัญหานี้ให้เป็นปัญหากำหนดการไม่เชิงเส้นที่ไม่มีเงื่อนไขบังคับได้ โดยใช้ lagrange multiplier w เพื่อกำจัดเงื่อนไขบังคับ $h(\mathbf{x})=0$ โดยที่ w เป็นตัวแปรแบบไม่จำกัดค่า ดังนั้นปัญหากำหนดการไม่เชิงเส้นนี้ จึงอยู่ในรูป

$$\text{minimize } f(\mathbf{x})+\mathbf{u}^{(k)t}\mathbf{g}(\mathbf{x})+wh(\mathbf{x})$$

ให้ $\mathbf{x}^{(k)}$ เป็นผลเฉลยเหมาะที่สุดของปัญหาย่อย

ขั้นที่ 3: ให้ $\theta^{(k)}=f(\mathbf{x}^{(k)})+\mathbf{u}^{(k)t}\mathbf{g}(\mathbf{x}^{(k)})$

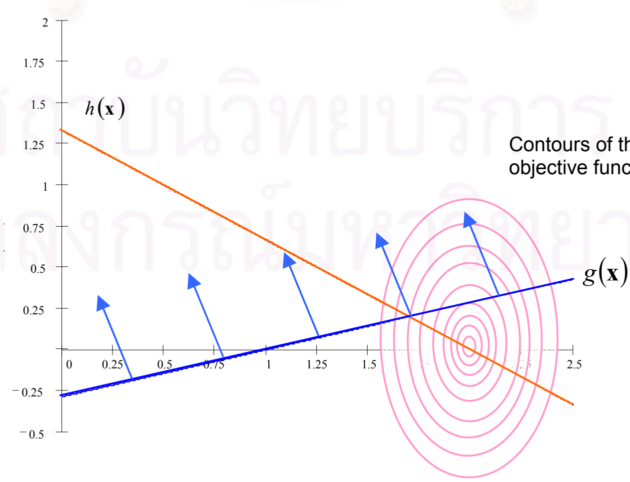
ขั้นที่ 4: ถ้า $\left|z^{(k)}-\theta^{(k)}\right|<\varepsilon$ แล้วจึงหยุดจะได้ $(\theta^{(k)},\mathbf{u}^{(k)})$ เป็นผลเฉลยเหมาะที่สุดของปัญหา lagrangian dual สำหรับกรณีที่ $\mathbf{x}\in\mathbf{D}_i$ เมื่อ ε เป็นจำนวนจริงบวกที่มีขนาดเล็ก ถ้าไม่เช่นนั้นให้เพิ่มเงื่อนไขบังคับ $z\le f(\mathbf{x}^{(k)})+\mathbf{u}^t\mathbf{g}(\mathbf{x}^{(k)})$ ในปัญหาหลัก เพิ่มค่า $k=k+1$ แล้วกลับไปขั้นตอนหลักอีกครั้ง

ตัวอย่างที่ 1

$$\text{minimize } (x_1-2)^2+\frac{1}{4}x_2^2$$

$$\text{subject to } x_1-\frac{7}{2}x_2-1\le 0$$

$$2x_1+3x_2=4$$



รูปที่ 3.1 แสดงถึงคอนทัวร์ของฟังก์ชันจุดประสงค์ตัดกับบริเวณที่เป็นไปได้ของตัวอย่างที่ 1

ในปัญหากำหนดการไม่เชิงเส้นนี้มี $\mathbf{D}_1 = \mathbf{D} = \{(x_1, x_2) \in \mathbf{R}^2\}$ เราให้

$$\mathbf{D}' = \{(x_1, x_2) : 2x_1 + 3x_2 = 4\}$$

ฟังก์ชัน lagrangian dual

$$\theta(u) = \min\{(x_1 - 2)^2 + \frac{1}{4}x_2^2 + u(x_1 - \frac{7}{2}x_2 - 1) : 2x_1 + 3x_2 = 4\}$$

<u>ขั้นตอนเริ่มต้น</u>

$$\mathbf{x}^{(0)} = \left(\frac{5}{4}, \frac{1}{2}\right), \varepsilon = 10^{-6}, k = 1$$

<u>ขั้นตอนหลัก</u>

รอบที่ 1

ขั้นที่ 1:

$$\begin{aligned} &\text{maximize } z \\ &\text{subject to} \quad z \le \frac{5}{8} - \frac{3}{2}u \\ &\qquad\qquad\quad u \ge 0 \end{aligned}$$

เมื่อ $f\left(\mathbf{x}^{(0)}\right) = \frac{5}{8}, g\left(\mathbf{x}^{(0)}\right) = -\frac{3}{2}$ ได้ $z^{(1)} = \frac{5}{8}, u^{(1)} = 0$ เป็นผลเฉลยที่เหมาะที่สุด

ขั้นที่ 2:

$$\begin{aligned} &\text{minimize} \quad (x_1 - 2)^2 + \frac{1}{4}x_2^2 \\ &\text{subject to} \quad 2x_1 + 3x_2 = 4 \end{aligned}$$

ได้ $\mathbf{x}^{(1)} = (2, 0)$ เป็นผลเฉลยที่เหมาะที่สุด

ขั้นที่ 3: $\theta^{(1)} = 0$

ขั้นที่ 4: $\theta^{(1)} < z^{(1)}$

รอบที่ 2

ขั้นที่ 1:

$$\begin{aligned} &\text{maximize } z \\ &\text{subject to} \quad z \le \frac{5}{8} - \frac{3}{2}u \\ &\qquad\qquad\quad z \le 0 + u \\ &\qquad\qquad\quad u \ge 0 \end{aligned}$$

เมื่อ $f\left(\mathbf{x}^{(1)}\right) = 0, g\left(\mathbf{x}^{(1)}\right) = 1$ ได้ $z^{(2)} = \frac{1}{4}, u^{(2)} = \frac{1}{4}$ เป็นผลเฉลยที่เหมาะที่สุด

ขั้นที่ 2:

$$\text{minimize} \quad (x_1-2)^2+\frac{1}{4}x_2^2+\frac{1}{4}(x_1-\frac{7}{2}x_2-1)$$

$$\text{subject to} \quad 2x_1+3x_2=4$$

ได้ $\mathbf{x}^{(2)}=\left(\frac{13}{8},\frac{1}{4}\right)$ เป็นผลเฉลยที่เหมาะที่สุด

ขั้นที่ 3: $\theta^{(2)}=\frac{3}{32}$

ขั้นที่ 4: $\theta^{(2)}<z^{(2)}$

รอบที่ 3

ขั้นที่ 1:

$$\text{maximize } z$$

$$\begin{aligned} \text{subject to} \quad & z \le \frac{5}{8}-\frac{3}{2}u \\ & z \le 0+u \\ & z \le \frac{5}{32}-\frac{1}{4}u \\ & u \ge 0 \end{aligned}$$

เมื่อ $f\left(\mathbf{x}^{(2)}\right)=\frac{5}{32}, g\left(\mathbf{x}^{(2)}\right)=-\frac{1}{4}$ ได้ $z^{(3)}=\frac{1}{8}, u^{(3)}=\frac{1}{8}$ เป็นผลเฉลยที่เหมาะที่สุด

ขั้นที่ 2:

$$\text{minimize} \quad (x_1-2)^2+\frac{1}{4}x_2^2+\frac{1}{8}(x_1-\frac{7}{2}x_2-1)$$

$$\text{subject to} \quad 2x_1+3x_2=4$$

ได้ $\mathbf{x}^{(3)}=\left(\frac{29}{16},\frac{1}{8}\right)$ เป็นผลเฉลยที่เหมาะที่สุด

ขั้นที่ 3: $\theta^{(3)}=\frac{11}{128}$

ขั้นที่ 4: $\theta^{(3)}<z^{(3)}$

ผลลัพธ์จาก 4 ขั้นตอนแรกเป็นตามตารางดังนี้

Iteration (k)	Constraint Added	$\left(z^{(k)},u^{(k)}\right)$	$\mathbf{x}^{(k)}$	$\theta^{(k)}$
1	$z\le\frac{5}{8}-\frac{3}{2}u$	$\left(\frac{5}{8},0\right)$	$(2,0)$	0
2	$z\le 0+u$	$\left(\frac{1}{4},\frac{1}{4}\right)$	$\left(\frac{13}{8},\frac{1}{4}\right)$	$\frac{3}{32}$
3	$z\le\frac{5}{32}-\frac{1}{4}u$	$\left(\frac{1}{8},\frac{1}{8}\right)$	$\left(\frac{29}{16},\frac{1}{8}\right)$	$\frac{11}{128}$
4	$z\le\frac{5}{128}+\frac{3}{8}u$	$\left(\frac{7}{64},\frac{3}{16}\right)$	$\left(\frac{55}{32},\frac{3}{16}\right)$	$\frac{51}{512}$

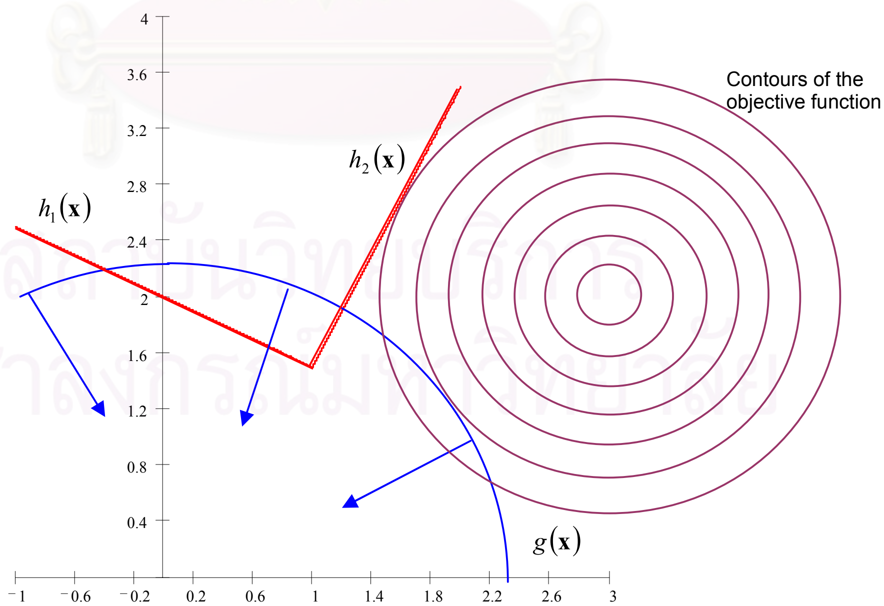
ฟังก์ชัน Lagrangian Dual มีค่าสูงสุดที่ $\bar{u}=\frac{1}{5}$ โดยที่ $\theta=\frac{1}{10}$

<u>ตัวอย่างที่ 2</u>

$$\begin{aligned}&\text{minimize} && (x_1-3)^2+(x_2-2)^2\\ &\text{subject to} && x_1^2+x_2^2-5\le 0\end{aligned}$$

$$h(\mathbf{x})=\begin{cases}x_1+2x_2-4=0;\mathbf{x}\in\mathbf{D}_1\\ 4x_1-2x_2-1=0;\mathbf{x}\in\mathbf{D}_2\end{cases}$$

เมื่อ $\mathbf{D}_1=\{x_1\,|\,-1\le x_1\le 1\}\times\mathbf{R}$ และ $\mathbf{D}_2=\{x_1\,|\,1\le x_1\le 2\}\times\mathbf{R}$



รูปที่ 3.2 แสดงถึงคอนทัวร์ของฟังก์ชันจุดประสงค์ตัดกับบริเวณที่เป็นไปได้ของตัวอย่างที่ 2

ขั้นที่ 1: แบ่งปัญหาออกเป็น 2 ส่วน

ส่วนที่ 1	ส่วนที่ 2
minimize $(x_1-3)^2+(x_2-2)^2$ subject to $x_1^2+x_2^2-5\le 0$ $-x_1-1\le 0$ $x_1-1\le 0$ $x_1+2x_2-4 \ =0$	minimize $(x_1-3)^2+(x_2-2)^2$ subject to $x_1^2+x_2^2-5\le 0$ $-x_1+1\le 0$ $x_1-2\le 0$ $4x_1-2x_2-1 \ =0$

ฟังก์ชัน lagrangian dual สำหรับส่วนที่ 1

$$\theta(u)=\min\{(x_1-3)^2+(x_1-2)^2+u_1(x_1^2+x_2^2\text{ - }5)+u_2(\text{-}x_1\text{ - }1)+u_3(x_1\text{ - }1):x_1\ +2x_2-4=0\}$$

ฟังก์ชัน lagrangian dual สำหรับส่วนที่ 2

$$\theta(u)=\min\{(x_1-3)^2+(x_1-2)^2+u_1(x_1^2+x_2^2\text{ - }5)+u_2(\text{-}x_1+1)+u_3(x_1\text{ - }2):4x_1\ -2x_2-1=0\}$$

ขั้นที่ 2: หาผลเฉลยของปัญหากำหนดการเชิงเส้นในช่วงที่ 1 $\mathbf{x}^{(0)}=(1,1.5)$

ผลลัพธ์จาก 4 ขั้นตอนแรกเป็นตามตารางดังนี้

Iteration (k)	Constraint Added	$\left(z^{(k)},u_1^{(k)},u_2^{(k)},u_3^{(k)}\right)$	$\mathbf{x}^{(k)t}$	$\theta^{(k)}$
1	$z\le 9-u_1-u_2-u_3$	$(9,0,0,0)$	$(2.4,0.8)$	1.8
2	$z\le 1.8+1.4u_1-3.4u_2+1.4u_3$	$(6,3,0,0)$	$(1.2,1.4)$	-1.2
3	$z\le 3.6-1.6u_1-2.2u_2+0.2u_3$	$(4.5,0,0,4.5)$	$(0.6,1.7)$	4.05
4	$z\le 5.85-1.75u_1+0.4u_2-0.3u_3$	$(4.35,0,0,3.75)$	$(0.9,1.55)$	4.237

ฟังก์ชัน lagrangian dual มีค่าสูงสุดที่ $\bar{\mathbf{u}}=(0,0,3.5)$ โดยที่ $\theta\ =4.25$

หาผลเฉลยของปัญหากำหนดการเชิงเส้นในช่วงที่ 2, $\mathbf{x}^{(0)}=(1\ ,1.5)$

ผลลัพธ์จาก 4 ขั้นตอนแรกเป็นตามตารางดังนี้

Iteration (k)	Constraint Added	$\left(z^{(k)},u_1^{(k)},u_2^{(k)},u_3^{(k)}\right)$	$\mathbf{x}^{(k)t}$	$\theta^{(k)}$
1	$z\le 4.25-1.75u_1-u_3$	$(4.25,0,0,0)$	$(1.6,2.7)$	2.45
2	$z\le 2.45+4.85u_1-0.6u_2-0.4u_3$	$(3.773,0.273,0,0)$	$(1.3,2.1)$	3.2
3	$z\le 2.9+1.1u_1-0.3u_2-0.7u_3$	$(3.421,0.474,0,0)$	$(0.6,1.7)$	3.255
4	$z\ \le 3.46-0.44u_1-0.15u_2-0.85u_3$	$(3.303,0.366,0,0)$	$(1.225,1.95)$	3.264

ฟังก์ชัน lagrangian dual มีค่าสูงสุดที่ $\bar{\mathbf{u}}=(0.397,0,0)$ โดยที่ $\theta\ =3.27$

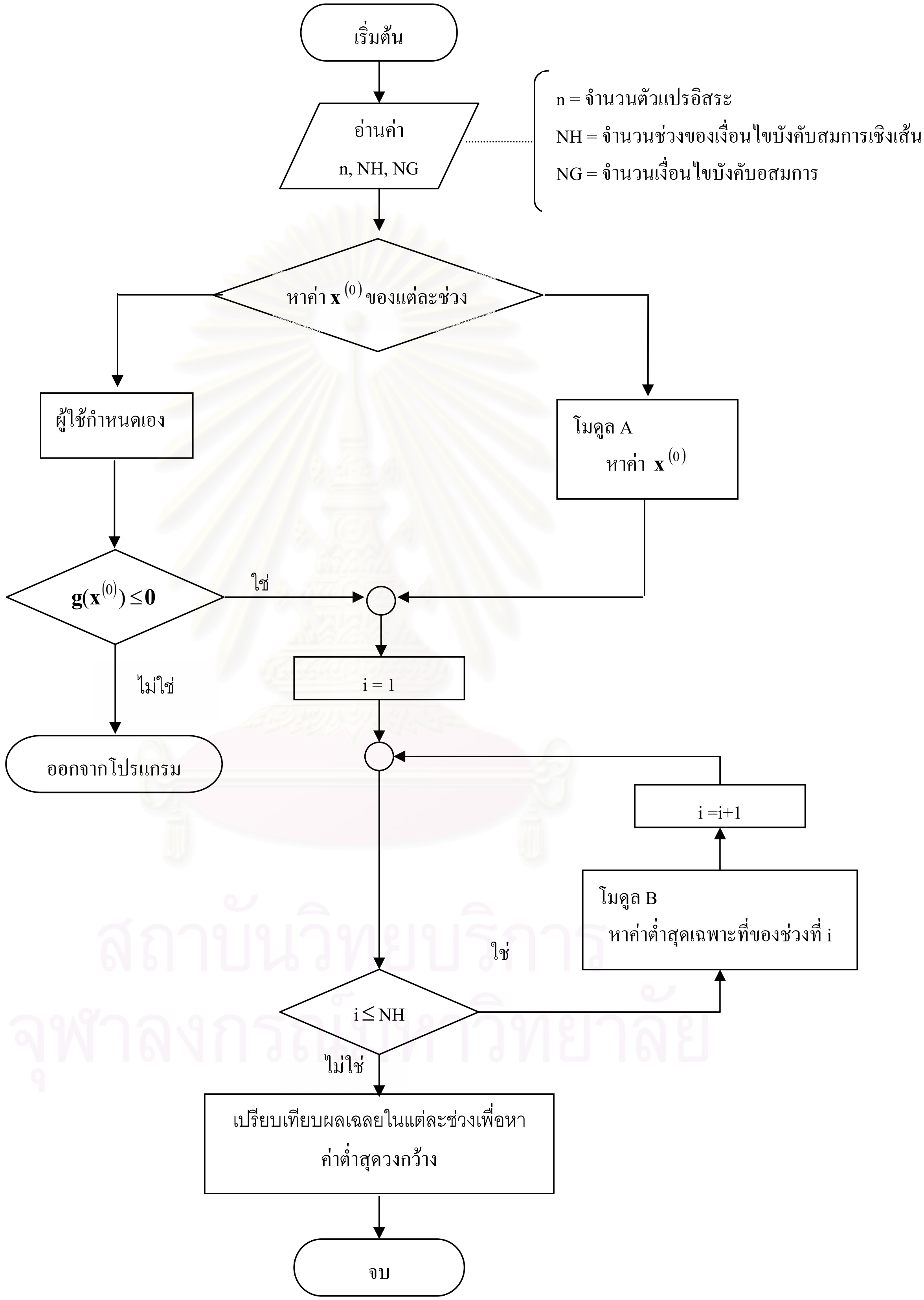
ขั้นที่ 3: เปรียบเทียบค่าต่ำสุดเฉพาะที่ของแต่ละช่วง

ค่าต่ำสุดเฉพาะที่จากช่วงที่ 1 เท่ากับ 4.25

ค่าต่ำสุดเฉพาะที่จากช่วงที่ 2 เท่ากับ 3.27

ดังนั้นค่าต่ำสุดวงกว้างเท่ากับ 3.27

3.3 ผังงาน


เริ่มต้น
อ่านค่า
n, NH, NG
n = จำนวนตัวแปรอิสระ
NH = จำนวนช่วงของเงื่อนไขบังคับสมการเชิงเส้น
NG = จำนวนเงื่อนไขบังคับอสมการ
หาค่า $\mathbf{x}^{(0)}$ ของแต่ละช่วง
ผู้ใช้กำหนดเอง
โมดูล A
หาค่า $\mathbf{x}^{(0)}$
$\mathbf{g}(\mathbf{x}^{(0)}) \leq \mathbf{0}$
ใช่
ไม่ใช่
ออกจากโปรแกรม
i = 1
i =i+1
โมดูล B
หาค่าต่ำสุดเฉพาะที่ของช่วงที่ i
ใช่
$i \leq NH$
ไม่ใช่
เปรียบเทียบผลเฉลยในแต่ละช่วงเพื่อหา
ค่าต่ำสุดวงกว้าง
จบ

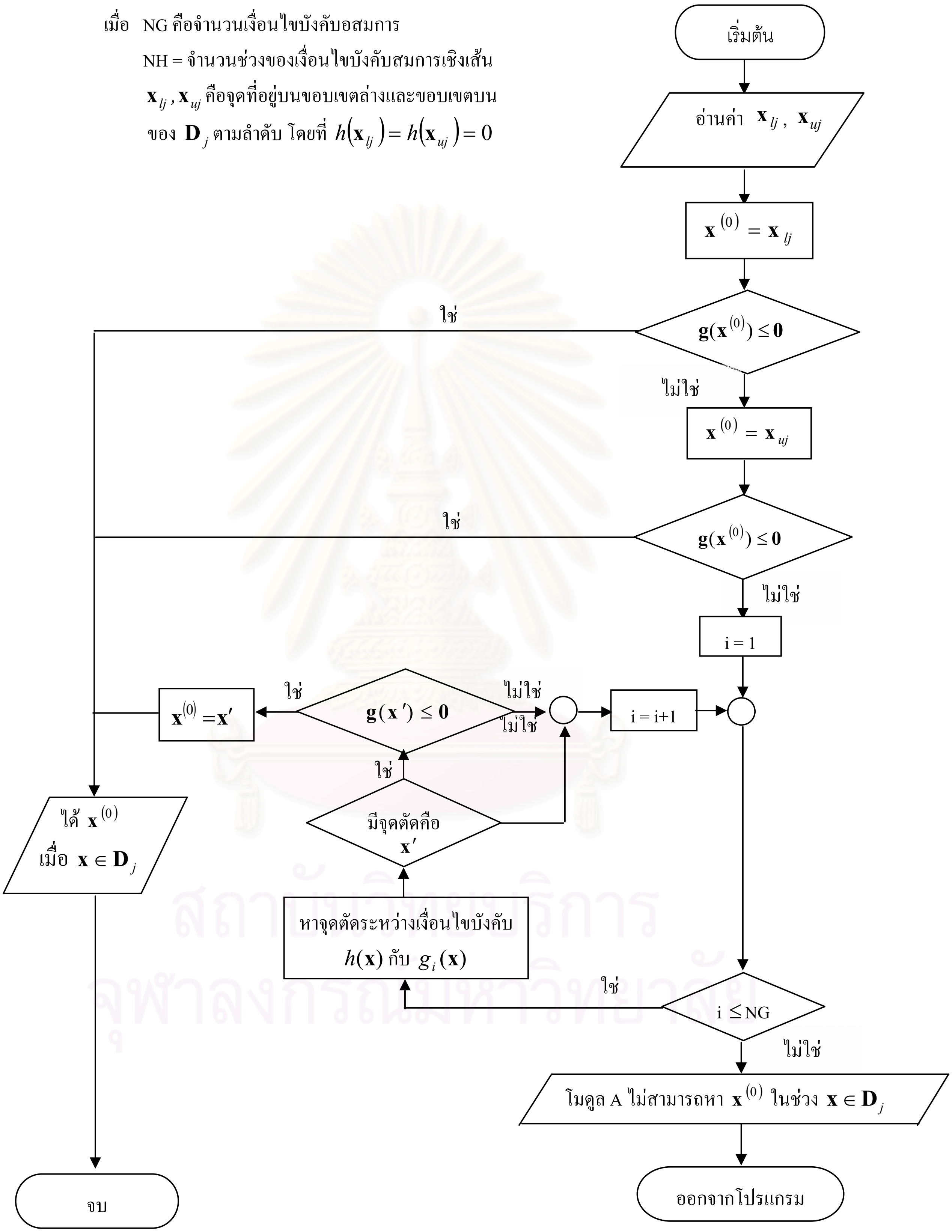
โมดูล A แผนผังการหาค่าเริ่มต้นเมื่อ $\mathbf{x} \in \mathbf{D}_j$ เมื่อ $j = 1,2,\ldots,\mathrm{NH}$

เมื่อ NG คือจำนวนเงื่อนไขบังคับอสมการ

NH = จำนวนช่วงของเงื่อนไขบังคับสมการเชิงเส้น

$\mathbf{x}_{lj}, \mathbf{x}_{uj}$ คือจุดที่อยู่บนขอบเขตล่างและขอบเขตบน

ของ $\mathbf{D}_j$ ตามลำดับ โดยที่ $h(\mathbf{x}_{lj}) = h(\mathbf{x}_{uj}) = 0$

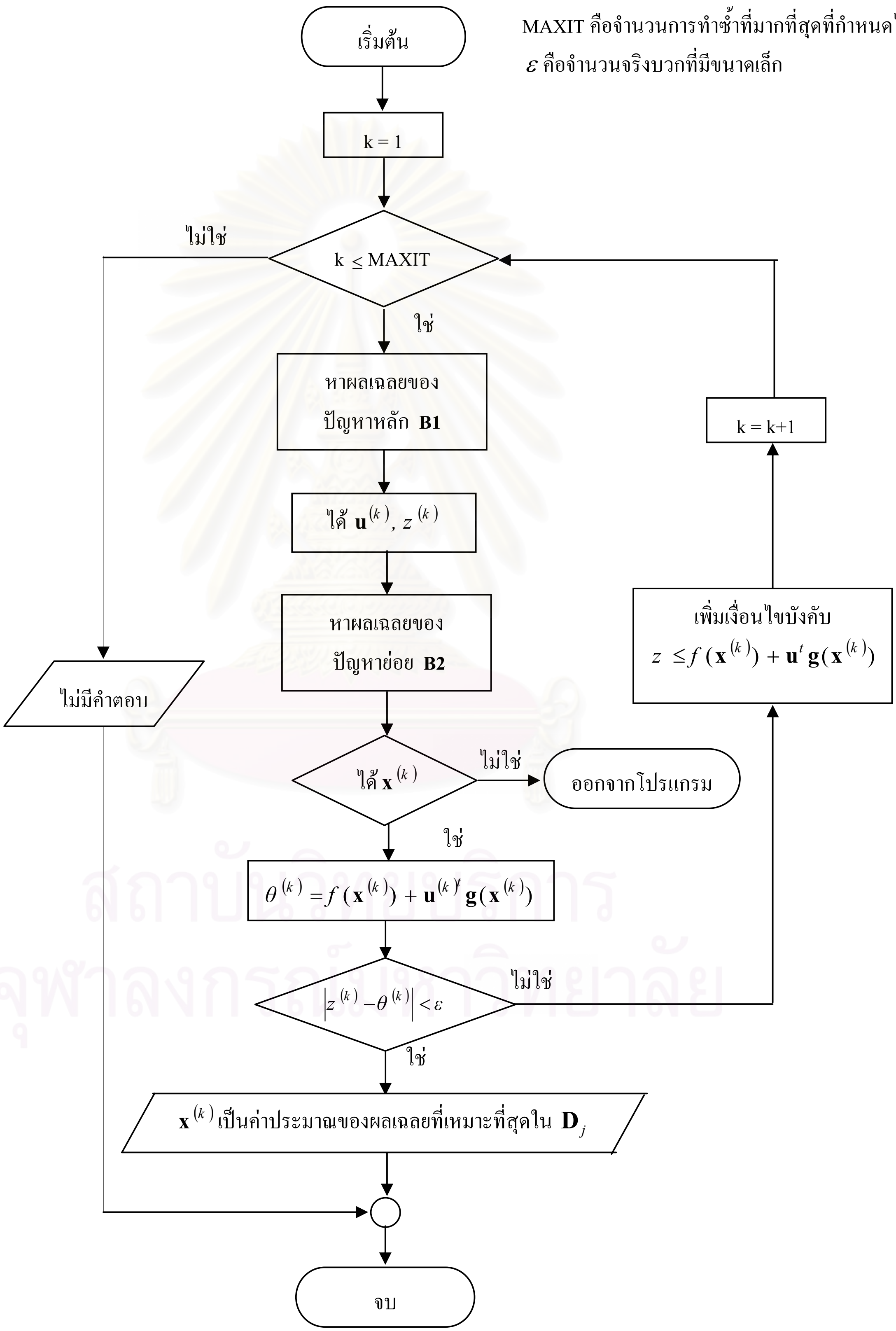


B แผนผังการหาค่าต่ำสุดเฉพาะที่เมื่อ $\mathbf{x} \in \mathbf{D}_j$ เมื่อ $j = 1,2,\ldots,\mathrm{NH}$

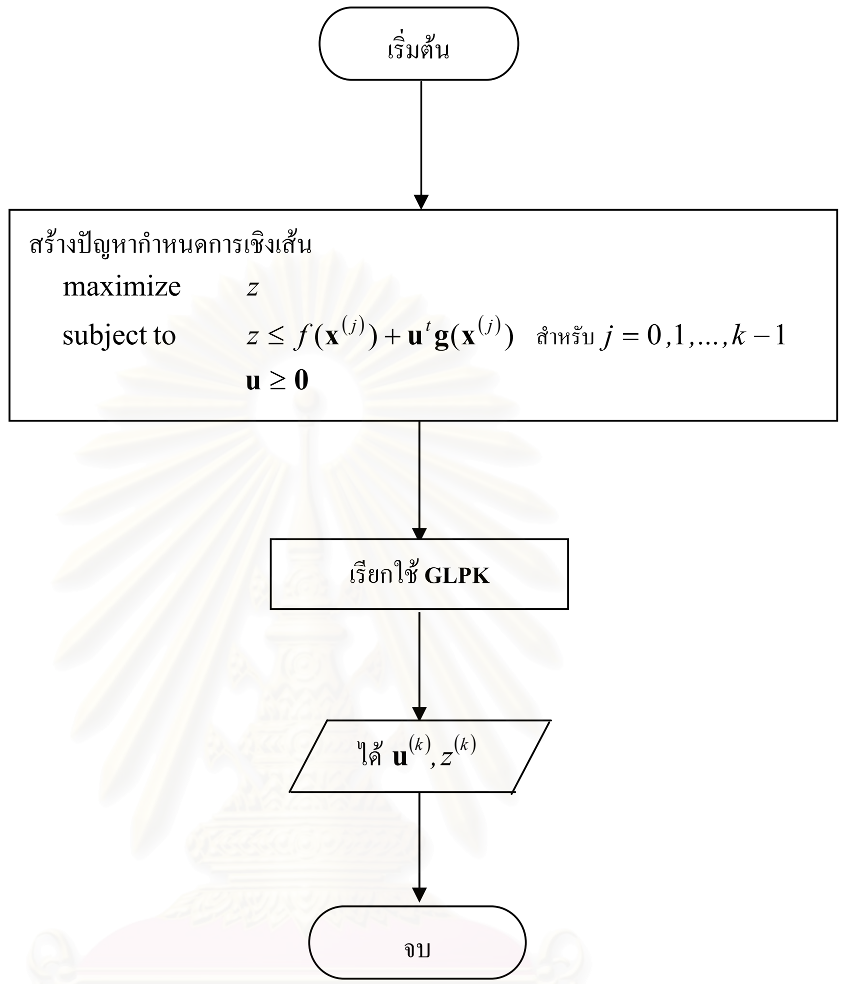
เมื่อ NH คือจำนวนช่วงของเงื่อนไขบังคับสมการเชิงเส้น

MAXIT คือจำนวนการทำซ้ำที่มากที่สุดที่กำหนดไว้

ε คือจำนวนจริงบวกที่มีขนาดเล็ก



B1 แผนผังการหาผลเฉลยของปัญหาหลัก



B2 แผนผังการหาผลเฉลยของปัญหาย่อย

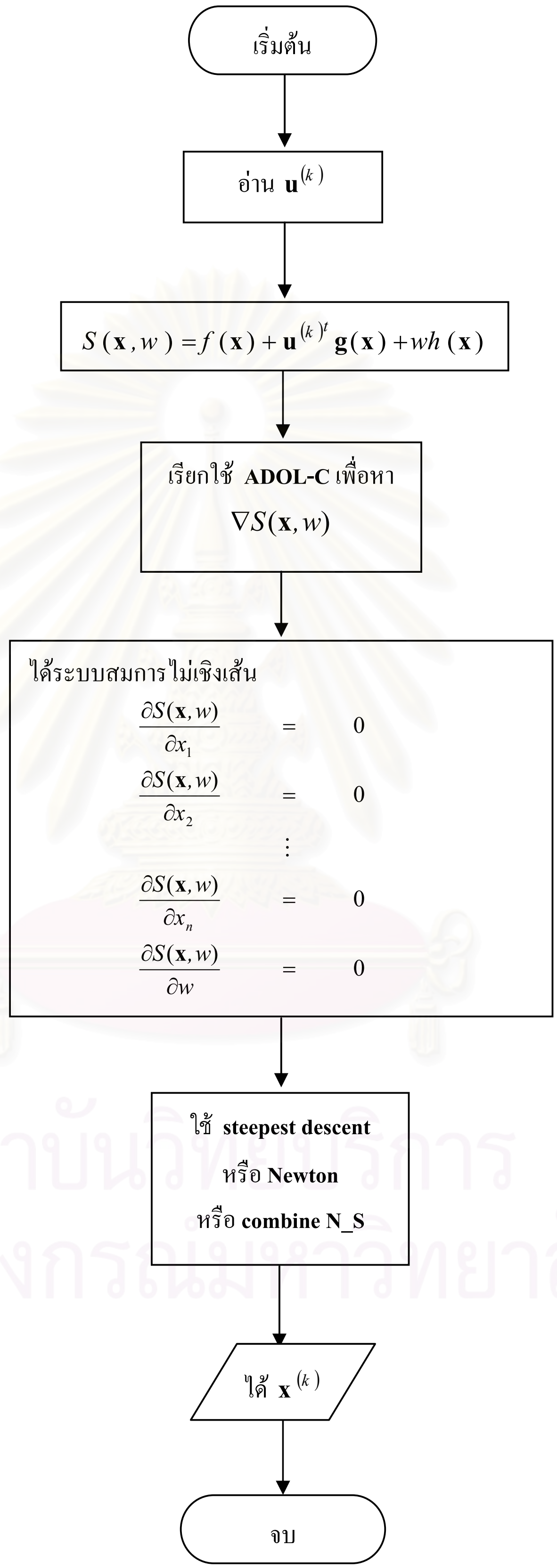
ปัญหาย่อยคือ

$$\text{minimize } S(\mathbf{x},w) = f(\mathbf{x}) + \mathbf{u}^{(k)t}\mathbf{g}(\mathbf{x}) + wh(\mathbf{x}) \text{ เมื่อ } \mathbf{x} \in \mathbf{D}_j \text{ สำหรับ } j = 1,2,\ldots,\mathrm{NH}$$

เมื่อ NH คือจำนวนช่วงของเงื่อนไขบังคับสมการเชิงเส้นปัญหาข้างต้นเป็นปัญหากำหนดการไม่เชิงเส้นแบบไม่มีเงื่อนไข และ $S(\mathbf{x},w)$ เป็นฟังก์ชันนูนต่อเนื่องที่สามารถหาอนุพันธ์ได้ จากความรู้ทางแคลคูลัสฟังก์ชัน $S(\mathbf{x},w)$ จะมีค่าต่ำสุดสัมพัทธ์เมื่อ $\nabla S(\mathbf{x},w) = \mathbf{0}$ นั่นคือ

$$\begin{aligned}
\frac{\partial S(\mathbf{x},w)}{\partial x_1} &= 0 \\
\frac{\partial S(\mathbf{x},w)}{\partial x_2} &= 0 \\
&\vdots \\
\frac{\partial S(\mathbf{x},w)}{\partial x_n} &= 0 \\
\frac{\partial S(\mathbf{x},w)}{\partial w} &= 0
\end{aligned}$$

ดังนั้นการหาผลเฉลยของปัญหาย่อยคือการแก้ปัญหาระบบสมการไม่เชิงเส้น สำหรับงานวิจัยนี้เลือกใช้วิธี Newton, วิธี steepest descent และวิธี combine N_S มาใช้ในการหาผลเฉลยของระบบสมการไม่เชิงเส้นซึ่งได้เขียนรายละเอียดของทั้ง 3 วิธีไว้ในภาคผนวก ก โดยผู้ใช้สามารถเลือกใช้วิธีใดวิธีหนึ่งได้ ถ้าผู้ใช้ไม่กำหนดโปรแกรมจะใช้วิธี combine N_S เราสามารถเขียนสรุปเป็นผังงานได้ดังนี้


เริ่มต้น
อ่าน $\mathbf{u}^{(k)}$
$S(\mathbf{x},w) = f(\mathbf{x}) + \mathbf{u}^{(k)^t}\mathbf{g}(\mathbf{x}) + wh(\mathbf{x})$
เรียกใช้ **ADOL-C** เพื่อหา $\nabla S(\mathbf{x},w)$
ได้ระบบสมการไม่เชิงเส้น
$\frac{\partial S(\mathbf{x},w)}{\partial x_1} = 0$
$\frac{\partial S(\mathbf{x},w)}{\partial x_2} = 0$
$\vdots$
$\frac{\partial S(\mathbf{x},w)}{\partial x_n} = 0$
$\frac{\partial S(\mathbf{x},w)}{\partial w} = 0$
ใช้ **steepest descent** หรือ **Newton** หรือ **combine N_S**
ได้ $\mathbf{x}^{(k)}$
จบ

บทที่ 4
การใช้งานซอฟต์แวร์ที่สร้างขึ้น

4.1 ขั้นตอนก่อนใช้งาน

ซอฟต์แวร์ CUT ที่สร้างขึ้นบีบอัดไฟล์ข้อมูลทั้งหมดอยู่ในไฟล์ CUT.tar.gz เริ่มต้นการใช้งานต้องคลายไฟล์ออกมาโดยใช้คำสั่ง

$gunzip CUT.tar.gz

คำสั่งนี้หมายถึง ให้คลายไฟล์ CUT.tar.gz เป็น CUT.tar จากนั้นใช้คำสั่ง

$tar –xvf CUT.tar

คำสั่งนี้หมายถึง คลายไฟล์ CUT.tar เป็น directory CUT ซึ่งประกอบไปด้วยไฟล์และกลุ่มไฟล์ที่อยู่ใน directory ดังนี้

1. Directory ประกอบด้วย 3 กลุ่มคือ

include	เป็น directory ที่ประกอบไปด้วย header file ของ GLPK
LIB	เป็น directory ที่ประกอบไปด้วย library file ของ GLPK คือ libglpk.a และ library file ของ ADOL-C คือ libad.a
SRC	เป็น directory ที่ประกอบไปด้วย header file ของ ADOL-C

2. File ประกอบด้วย 6 ไฟล์คือ

adol.c	เป็นไฟล์ที่ใช้ในการคำนวณหาค่า gradient และ Hessian ของฟังก์ชันจุดประสงค์ในปัญหาย่อย
tag.c	เป็นไฟล์ที่ใช้ในการคำนวณหาค่า gradient ของเงื่อนไขบังคับของปัญหากำหนดการไม่เชิงเส้นทั้งที่เป็นอสมการไม่เชิงเส้น และสมการเชิงเส้นเป็นช่วง
initial.c	เป็นไฟล์ที่ใช้หาค่าเริ่มต้นให้กับปัญหา
cut.c	เป็นไฟล์หลักควบคุมการทำงานตามวิธีระนาบตัด
prob.txt	เป็นไฟล์ข้อมูลของปัญหา
makefile	เป็นไฟล์ที่สั่งให้คอมพิวเตอร์แปลโปรแกรมเป็นภาษาเครื่องคอมพิวเตอร์

4.2 ขั้นตอนการใช้งาน

เริ่มต้นการใช้งาน ผู้ใช้งานต้องเตรียมไฟล์ข้อมูล prob.txt และกำหนดฟังก์ชันต่าง ๆ ใน file adol.c, tag.c, initial.c และ cut.c แล้วพิมพ์คำสั่ง

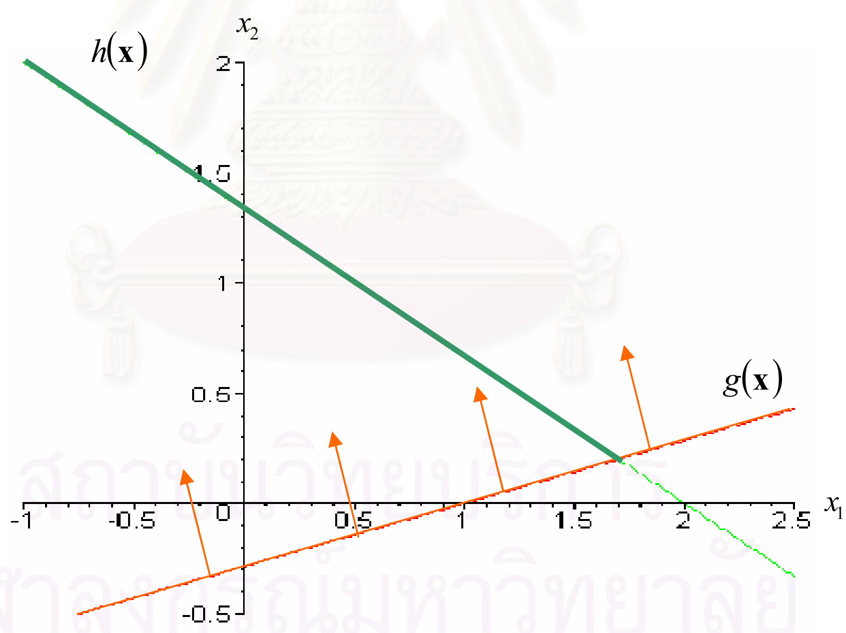
```
$makefile
```

คอมพิวเตอร์แปลโปรแกรมเป็นภาษาเครื่อง เราสามารถดูผลลัพธ์ที่ได้จากการพิมพ์คำสั่ง

```
$./cut
```

<u>**ตัวอย่างที่1**</u>

$$\begin{aligned} \text{minimize} \quad & (x_1-2)^2+\frac{1}{4}x_2^2 \\ \text{subject to} \quad & x_1-\frac{7}{2}x_2-1\le 0 \\ & 2x_1+3x_2-4=0 \end{aligned}$$

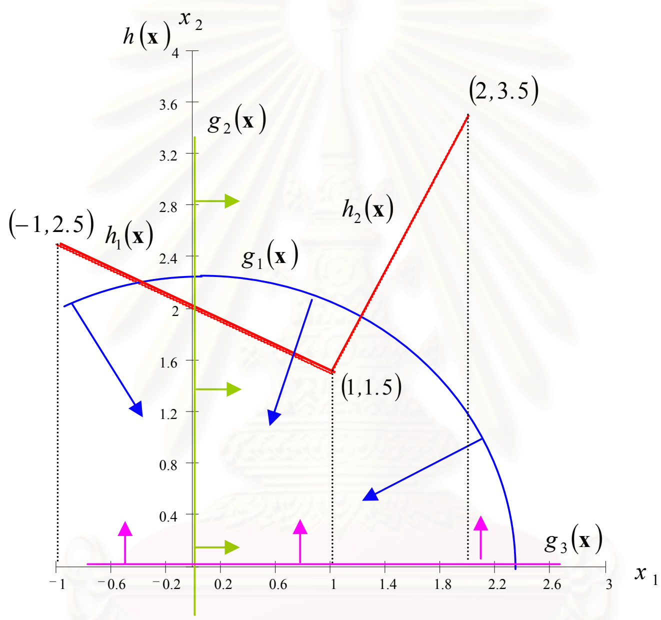


รูปที่ 4.1 แสดงถึงบริเวณที่เป็นไปได้และโดเมน **x** ของตัวอย่างที่ 1

<u>ตัวอย่างที่2</u>

$$\text{minimize } (x_1-3)^2+(x_2-2)^2$$
$$\text{subject to} \quad x_1^2+x_2^2-5\le 0$$
$$-x_1\le 0$$
$$-x_2\le 0$$
$$h(\mathbf{x})=\begin{cases} x_1+2x_2-4=0 & \mathbf{x}\in\mathbf{D}_1 \\ 4x_1-2x_2-1=0 & \mathbf{x}\in\mathbf{D}_2 \end{cases}$$

เมื่อ $\mathbf{D}_1=\{x_1 \mid -1\le x_1\le 1\}\times\mathbf{R}$ และ $\mathbf{D}_2=\{x_1 \mid 1\le x_1\le 2\}\times\mathbf{R}$



รูปที่ 4.2 แสดงถึงบริเวณที่เป็นไปได้และโดเมน **x** ของตัวอย่างที่ 2

ไฟล์ prob.txt เป็นไฟล์ที่ใช้บันทึกรายละเอียดของปัญหาคือ วิธีที่ใช้สำหรับแก้ปัญหาระบบสมการไม่เชิงเส้น (1= วิธี Newton, 2=วิธี steepest descent, 3=วิธี combine N_S) จำนวนตัวแปรอิสระ จำนวนเงื่อนไขบังคับอสมการ และจำนวนช่วงของเงื่อนไขบังคับสมการเชิงเส้น หากสมการเชิงเส้นมีมากกว่า 1ช่วง จะต้องบอกถึงจุดที่เป็นจุดเริ่มต้นและจุดปลายของแต่ละช่วง เพื่อช่วยให้การหาค่า $\mathbf{x}^{(0)}$ สะดวกและมีความถูกต้องมากขึ้น

ตัวอย่างที่ 1	ตัวอย่างที่ 2
1 // วิธีแก้ปัญหาระบบสมการไม่เชิงเส้น	1 // วิธีแก้ปัญหาระบบสมการไม่เชิงเส้น
2 //จำนวนตัวแปรอิสระ	2 //จำนวนตัวแปรอิสระ
1 //จำนวนเงื่อนไขบังคับอสมการ	3 //จำนวนเงื่อนไขบังคับอสมการ
1 //จำนวนช่วงของเงื่อนไขบังคับสมการเชิงเส้น	3 //จำนวนช่วงของเงื่อนไขบังคับสมการเชิงเส้น
	-1 2.5// (x_1, x_2) ที่เป็นจุดเริ่มต้นของ $h_1(\mathbf{x})$
	1 1.5// (x_1, x_2) ที่เป็นจุดปลายของ $h_1(\mathbf{x})$
	2 3.5// (x_1, x_2) ที่เป็นจุดปลายของ $h_2(\mathbf{x})$

ไฟล์ tag.c มีการเปลี่ยนฟังก์ชันย่อย 2 ฟังก์ชันคือ

(1) adouble GX(int i, adouble xx[]) เป็นฟังก์ชันที่ใช้คำนวณเงื่อนไขบังคับ **g(xx)** ของปัญหากำหนดการไม่เชิงเส้น โดยมีข้อมูลเข้า 2 ตัวคือ

- ข้อมูลเข้า int i มีชนิดตัวแปรเป็นจำนวนเต็ม เป็นค่าที่แสดงให้รู้ว่าเป็นการคำนวณจากฟังก์ชัน g ตัวไหน เช่นถ้า i=0 แสดงว่าเป็นการคำนวณจากฟังก์ชัน g_1
- ข้อมูลเข้า adouble xx[] มีชนิดตัวแปรเป็น adouble ซึ่งเป็นตัวแปรในโปรแกรม ADOL-C **xx** จะรับค่าจาก **x** ที่อยู่ในโปรแกรมหลัก

ข้อมูลออกที่ได้คือค่าที่ได้จากการคำนวณ g_i(**xx**)

```
/* ตัวอย่างที่ 1 */
adouble GX(int i, adouble xx[])
{       adouble gg;
        switch(i+1)
        {       case 1:   gg = xx[0]-7.0/2.0*xx[1]-1; break;
        }
        return gg;
}
```

```
/* ตัวอย่างที่ 2 */
adouble GX(int i, adouble xx[])
{       adouble gg;
        switch(i+1)
        {       case 1:   gg = xx[0]* xx[0]+ xx[1]* xx[1]-5; break;
                case 2 :  gg = -xx[0]; break;
                case 3:   gg = -xx[1]; break;
        }
        return gg;
}
```

(2) adouble HX(int pcs, adouble xx[]) เป็นฟังก์ชันที่ใช้คำนวณเงื่อนไขบังคับ $h(\mathbf{xx})$ ของปัญหากำหนดการไม่เชิงเส้น โดยมีข้อมูลเข้า 2 ตัวคือ

- ข้อมูลเข้า int pcs มีชนิดตัวแปรเป็นจำนวนเต็ม เป็นค่าที่แสดงให้รู้ว่า **xx** อยู่ใน **D** ใด ถ้า pcs=0 แสดงว่า $\mathbf{xx} \in \mathbf{D}_1$
- ข้อมูลเข้า adouble xx[] มีชนิดตัวแปรเป็น adouble ซึ่งเป็นตัวแปรในโปรแกรม ADOL-C **xx** จะรับค่าจาก **x** ที่อยู่ในโปรแกรมหลัก

ข้อมูลออกที่ได้คือค่าที่ได้จากการคำนวณ $h(\mathbf{xx})$ เมื่อ $\mathbf{xx} \in \mathbf{D}_{pcs}$

```
/* ตัวอย่างที่ 1 */
adouble HX(int pcs, adouble xx[])
{       adouble hh;
        switch(pcs+1)
        {       case 1:   hh = 2*xx[0]+3*xx[1]-4;  break;
        }
        return hh;
}
```

```
/* ตัวอย่างที่ 2 */
adouble HX(int pcs, adouble xx[])
{       adouble hh;
        switch(pcs+1)
```

```
    {       case 1:  hh = xx[0]+2*xx[1]-4;  break;
            case 2:  hh = 4*xx[0]-2*xx[1]-1;  break;
    }
    return hh;
}
```

ไฟล์ initial.c มีการเปลี่ยนฟังก์ชันย่อย 2 ฟังก์ชันคือ

(1) double GX(int i, double xx[]) เป็นฟังก์ชันที่ใช้คำนวณเงื่อนไขบังคับ **g(xx)** ของปัญหากำหนดการไม่เชิงเส้น โดยมีข้อมูลเข้า 2 ตัวคือ

- ข้อมูลเข้า int i มีชนิดตัวแปรเป็นจำนวนเต็ม เป็นค่าที่แสดงให้รู้ว่าเป็นการคำนวณจากฟังก์ชัน g ตัวไหน เช่นถ้า i=0 แสดงว่าเป็นการคำนวณจากฟังก์ชัน g_1
- ข้อมูลเข้า double xx[] มีชนิดตัวแปรเป็น double **xx** จะรับค่าจาก **x** ที่อยู่ในโปรแกรมหลัก

ข้อมูลออกที่ได้คือค่าที่ได้จากการคำนวณ g_i(**xx**)

```
/* ตัวอย่างที่ 1 */
double GX(int i,double *xx)
{       double gg;
        switch(i+1)
        {       case 1: gg = xx[0]-7.0/2.0*xx[1]-1; break;
        }
        return gg;
}
```

```
/* ตัวอย่างที่ 2 */
double GX(int i,double *xx)
{       double gg;
        switch(i+1)
        {       case 1: gg = xx[0]* xx[0]+ xx[1]* xx[1]-5; break;
                case 2: gg = -xx[0]; break;
                case 3: gg = -xx[1]; break;
        }
        return gg;
```

```
}
```

(2) ฟังก์ชัน double HX(int pcs, double xx[]) เป็นฟังก์ชันที่ใช้คำนวณเงื่อนไขบังคับ $h(\mathbf{xx})$ ของปัญหากำหนดการไม่เชิงเส้น โดยมีข้อมูลเข้า 2 ตัวคือ

- ข้อมูลเข้า int pcs มีชนิดตัวแปรเป็นจำนวนเต็ม เป็นค่าที่แสดงให้รู้ว่า **xx** อยู่ใน **D** ใด ถ้า pcs=0 แสดงว่า $\mathbf{xx} \in \mathbf{D}_1$
- ข้อมูลเข้า double xx[] มีชนิดตัวแปรเป็น double **xx** จะรับค่าจาก **x** ที่อยู่ในโปรแกรมหลัก

ข้อมูลออกที่ได้คือค่าที่ได้จากการคำนวณ $h(\mathbf{xx})$ เมื่อ $\mathbf{xx} \in \mathbf{D}_{pcs}$

```
/* ตัวอย่างที่ 1 */
double HX(int pcs, double *xx)
{	double hh;
	switch(pcs+1)
	{	case 1:   hh = 2*xx[0]+3*xx[1]-4;  break;
	}
	return hh;
}
```

```
/* ตัวอย่างที่ 2 */
double HX(int pcs, double *xx)
{	double hh;
	switch(pcs+1)
	{	case 1:  hh = xx[0]+2*xx[1]-4;  break;
		case 2:  hh = 4*xx[0]-2*xx[1]-1;  break;
	}
	return hh;
}
```

ไฟล์ adol.c มีการเปลี่ยนฟังก์ชันย่อย 3 ฟังก์ชันคือ

(1) ฟังก์ชัน adouble FX(adouble xx []) เป็นฟังก์ชันที่ใช้คำนวณค่าจุดประสงค์ มีข้อมูลเข้าคือ **xx** มีชนิดตัวแปรเป็น adouble ซึ่งเป็นตัวแปรในโปรแกรม ADOL-C โดยที่ **xx** จะรับค่าจาก **x** ที่อยู่ในโปรแกรม หลัก ข้อมูลออกที่ได้คือค่าที่ได้จากการคำนวณ $f(\mathbf{xx})$

```
/* ตัวอย่างที่ 1 */
adouble FX( adouble xx[])
{         adouble f;
          f = (xx[0]-2)*( xx[0]-2)+0.25*xx[1]* xx[1]; break;
          return f;
}
```

```
/* ตัวอย่างที่ 2 */
adouble FX( adouble xx[])
{         adouble f;
          f = (xx[0]-3)*( xx[0]-3)+( xx[1]-2)*( xx[1]-2); break;
          return f;
}
```

(2) ฟังก์ชัน adouble GX(int pcs,int i, adouble xx[]) เป็นฟังก์ชันที่ใช้คำนวณเงื่อนไขบังคับ **g(xx)** ของปัญหากำหนดการไม่เชิงเส้น จะเห็นได้ว่าฟังก์ชัน GX ในไฟล์ **adol.c** แตกต่างจากไฟล์ **tag.c** และ **initial.c** ที่มีข้อมูลเข้าถึง 3 ตัวคือ

- ข้อมูลเข้า int pcs มีชนิดตัวแปรเป็นจำนวนเต็ม เป็นค่าที่แสดงให้รู้ว่า **xx** อยู่ใน **D** ใด ถ้า pcs=0 แสดงว่า $\mathbf{xx} \in \mathbf{D}_1$
- ข้อมูลเข้า int i มีชนิดตัวแปรเป็นจำนวนเต็ม เป็นค่าที่แสดงให้รู้ว่าเป็นการคำนวณจากฟังก์ชัน g ตัวไหน แต่จะแตกต่างกับไฟล์ **tag.c** และ **initial.c** ที่ว่า **g(xx)** ในไฟล์ **adol.c** จะเพิ่มเงื่อนไขบังคับว่า **xx** จะต้องอยู่ใน $\mathbf{D}_{pcs}$
- ข้อมูลเข้า adouble xx[] มีชนิดตัวแปรเป็น adouble ซึ่งเป็นตัวแปรในโปรแกรม ADOL-C **xx** จะรับค่าจาก **x** ที่อยู่ในโปรแกรมหลัก

ข้อมูลออกที่ได้คือค่าที่ได้จากการคำนวณ $g_i(\mathbf{xx})$

```
/* ตัวอย่างที่ 1 */
adouble GX(int pcs,int i, adouble xx[])
{         adouble gg;
          switch(pcs+1)
          {         case 1:   switch(i+1)
                              {         case 1: gg = gg = xx[0]-7.0/2.0*xx[1]-1; break;
```

```
                }
                break;
        }
        return gg;
}
```

```
/* ตัวอย่างที่ 2 */
adouble GX(int pcs,int i, adouble xx[])
{       adouble gg;
        switch(pcs+1)
        {       case 1:  switch(i+1)
                        {       case 1: gg = xx[0]* xx[0]+ xx[1]* xx[1]-5; break;
                                case 2: gg = -xx[0]; break;
                                case 3: gg = -xx[1]; break;
                                case 4: gg = -xx[0]-1; break; //lower bound D1
                                case 5: gg = xx[0]-1; break; // upper bound D1
                        }
                        break;
                case 2:  switch(i+1)
                        {       case 1: gg = xx[0]* xx[0]+ xx[1]* xx[1]-5; break;
                                case 2: gg = -xx[0]; break;
                                case 3: gg = -xx[1]; break;
                                case 4: gg = -xx[0]+1; break; //lower bound D2
                                case 5: gg = xx[0]-2; break;   //upper bound D2
                        }
                         break;
        }
        return gg;
}
```

(3) ฟังก์ชัน adouble HX(int pcs, adouble xx[]) เป็นฟังก์ชันที่ใช้คำนวณเงื่อนไขบังคับ $h(\mathbf{xx})$ ของปัญหากำหนดการไม่เชิงเส้น โดยมีข้อมูลเข้า 2 ตัวคือ

- ข้อมูลเข้า int pcs มีชนิดตัวแปรเป็นจำนวนเต็ม เป็นค่าที่แสดงให้รู้ว่า **xx** อยู่ใน **D** ใด ถ้า pcs=0 แสดงว่า $\mathbf{xx} \in \mathbf{D}_1$
- ข้อมูลเข้า adouble xx[] มีชนิดตัวแปรเป็น adouble ซึ่งเป็นตัวแปรในโปรแกรม ADOL-C **xx** จะรับค่าจาก **x** ที่อยู่ในโปรแกรมหลัก

ข้อมูลออกที่ได้คือค่าที่ได้จากการคำนวณ $h(\mathbf{xx})$ เมื่อ $\mathbf{xx} \in \mathbf{D}_{pcs}$

```
/* ตัวอย่างที่ 1 */
adouble HX(int pcs,adouble xx[])
{       adouble hh;
        switch(pcs+1)
        {       case 1: hh = 2*xx[0]+3*xx[1]-4;  break;
        }
        return hh;
}
```

```
/* ตัวอย่างที่ 2 */
adouble HX(int pcs,adouble xx[])
{       adouble hh;
        switch(pcs+1)
        {       case 1: hh = xx[0]+2*xx[1]-4;  break;
                case 2: hh = 4*xx[0]-2*xx[1]-1;  break;
        }
        return hh;
}
```

ไฟล์ cut.c มีการเปลี่ยนฟังก์ชันย่อย 2 ฟังก์ชันคือ

(1) ฟังก์ชัน double FX(double x[]) เป็นฟังก์ชันที่ใช้คำนวณค่าจุดประสงค์ มีข้อมูลเข้าคือ **x** ซึ่งเป็นตัวแปรชนิด double ข้อมูลออกที่ได้คือค่าที่ได้จากการคำนวณ $f(\mathbf{x})$

```
/* ตัวอย่างที่ 1 */
double FX( double x[])
{       double f;
        f = (x[0]-2)*(x[0]-2)+0.25*x[1]*x[1]; break;
        return f;
}
```

```
/* ตัวอย่างที่ 2 */
double FX( double x[])
{       double f;
        f = (x[0]-3)*(x[0]-3)+(x[1]-2)*(x[1]-2);  break;
        return f;
}
```

(2) ฟังก์ชัน double GX(int pcs,int i, double x[]) เป็นฟังก์ชันที่ใช้คำนวณเงื่อนไขบังคับ **g(x)** ของปัญหากำหนดการไม่เชิงเส้น จะเห็นได้ว่าฟังก์ชัน GX ในไฟล์ **cut.c** แตกต่างจากไฟล์ **tag.c** และ **initial.c** ที่มีข้อมูลเข้าถึง 3 ตัว และแตกต่างจากไฟล์ **adol.c** ที่ข้อมูลเข้าของ **x** เป็นตัวแปรชนิด double และข้อมูลออกที่ได้ก็เป็นชนิด double ที่เป็นเช่นนั้นเพราะว่า ฟังก์ชัน GX ที่อยู่ในไฟล์ cut.c เป็นโปรแกรมหลัก แต่ฟังก์ชัน GX ในไฟล์ adol.c เป็นฟังก์ชันที่ใช้ในโปรแกรม ADOL-C ดังนั้นชนิดของตัวแปรจึงเป็น adouble

```
/* ตัวอย่างที่ 1 */
double GX(int pcs,int i, double x[])
{       double gg;
        switch(pcs+1)
        {       case 1:  switch(i+1)
                         {       case 1: gg = x[0]-7.0/2.0*x[1]-1; break;
                         }
                         break;
        }
        return gg;
}
```

```
/* ตัวอย่างที่ 2 */
double GX(int pcs,int i, double x[])
{	double gg;
	switch(pcs+1)
	{	case 1:	switch(i+1)
			{	case 1: gg = x[0]*x[0]+x[1]*x[1]-5; break;
				case 2: gg = -x[0]; break;
				case 3: gg = -x[1]; break;
				case 4: gg = -x[0]-1; break; //lower bound D1
				case 5: gg = x[0]-1; break; // upper bound D1
			}
			break;
		case 2:	switch(i+1)
			{	case 1: gg = x[0]*x[0]+x[1]*x[1]-5; break;
				case 2: gg = -x[0]; break;
				case 3: gg = -x[1]; break;
				case 4: gg = -x[0]+1; break; //lower bound D2
				case 5: gg = x[0]-2; break;  //upper bound D2
			}
			break;
	}
	return gg;
}
```

บทที่ 5
การทดสอบซอฟต์แวร์และบทสรุป

5.1 การทดสอบผลการวิจัย

การทดสอบซอฟต์แวร์สำหรับหาผลเฉลยของปัญหากำหนดการไม่เชิงเส้นภายใต้เงื่อนไขบังคับอสมการไม่เชิงเส้น และสมการเชิงเส้นเป็นช่วง ได้ทำการทดสอบกับปัญหาทั้งหมด 10 ตัวอย่าง โดยให้ $\varepsilon = 0.0001$ ซึ่งได้แสดงผลลัพธ์จากการประมวลผลทั้งหมด และคำนวณเวลาที่ใช้ในการประมวลผลดังรูปที่ 5-1

```
clock_t  start,end;
double time;
start = clock();
.
.
.
end = clock();
time = ((double)(end-start))/CLOCKS_PER_SEC;
```

รูปที่ 5-1 แสดงการคำนวณหาเวลาในการประมวลผล

ปัญหา NLP1

$$
\begin{array}{lll}
\text{mininmize} & (x_1-2)^2+\frac{1}{4}x_2^2 & \\
\text{subject to} & x_1-\frac{7}{2}x_2-1 & \le 0 \\
 & 2x_1+3x_2-4 & = 0
\end{array}
$$

ค่าเริ่มต้นที่ได้จากโปรแกรมคือ $\mathbf{x}^{(0)} = (1.699967, 0.199996)^t$

จากการหาผลเฉลยโดยใช้ซอฟต์แวร์ที่สร้างขึ้นกับ GAMS สามารถเปรียบเทียบผลการคำนวณได้ดังนี้

Software		Optimal solution	Objective value	Iteration		Time
				Cutting	Total	
CUT	NEWTON	(1.709375 , 0.193750)	0.099902	6	12	0.03
	Steepest descent	(1.709379 , 0.193748)	0.099902	6	267	1.14
	Combine N_S	(1.709375 , 0.193750)	0.099902	6	18	0.04
GAMS		(1.7 , 0.2)	0.1	1		0.11

ปัญหา NLP2

$$
\begin{aligned}
\text{minimize} \quad & e^{x_1} + e^{x_2} + x_1 + 2x_2^2 + x_1^2 \\
\text{subject to} \quad & e^{x_1} + 2x_2 \le 6 \\
& -x_1 \le 0 \\
& -x_2 \le 0 \\
& x_1 - x_2 = 0
\end{aligned}
$$

ค่าเริ่มต้นที่ได้จากโปรแกรมคือ $\mathbf{x}^{(0)} = (1.251758, 1.251759)^t$

จากการหาผลเฉลยโดยใช้ซอฟต์แวร์ที่สร้างขึ้นกับ GAMS สามารถเปรียบเทียบผลการคำนวณได้ดังนี้

Software		Optimal solution	Objective value	Iteration		Time
				Cutting	Total	
Cut	Newton	(0.002734 , 0.002734)	1.99997	10	44	0.08
	Steepest descent	(0.002667 , 0.002672)	1.999962	10	5817	24.72
	Combine N_S	(0.002734 , 0.002734)	1.99997	10	57	0.08
GAMS		(0.000 , 0.000)	2	1		0.11

ปัญหา NLP3

$$
\begin{aligned}
\text{mininmize} \quad & 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \\
\text{subject to} \quad & 2x_1^2 - x_2 \le 0 \\
& -x_1 \le 0 \\
& -x_2 \le 0 \\
& x_1 + 5x_2 - 5 = 0
\end{aligned}
$$

ค่าเริ่มต้นที่ได้จากโปรแกรมคือ $\mathbf{x}^{(0)} = (0.000000, 0.999998)^t$

จากการหาผลเฉลยโดยใช้ซอฟต์แวร์ที่สร้างขึ้นกับ GAMS สามารถเปรียบเทียบผลการคำนวณได้ดังนี้

Software		Optimal solution	Objective value	Iteration		Time
				Cutting	Total	
Cut	Newton	(0.662608 , 0.867478)	-6.613143	8	14	0.03
	Steepest descent	(0.662599 , 0.867480)	-6.613143	8	173	0.82
	Combine N_S	(0.662609 , 0.867478)	-6.613143	8	23	0.07
GAMS		(0.659 , 0.868)	-6.613143	8		0.11

ปัญหา NLP4

$$
\begin{aligned}
\text{minimize} \quad & e^{x_1} + x_1^2 + 4x_1 + 2x_2^2 - 6x_2 + 2x_3 \\
\text{subject to} \quad & x_1^2 + e^{x_2} + 6x_3 \le 40 \\
& x_1 - x_2 + 5x_3 - 25 = 0 \\
& 0 \le x_1 \le 4 \\
& 0 \le x_2 \le 2 \\
& 0 \le x_3
\end{aligned}
$$

ค่าเริ่มต้นที่ได้จากโปรแกรมคือ $\mathbf{x}^{(0)} = (0.961538, 0.000000, 4.807692)^t$

จากการหาผลเฉลยโดยใช้ซอฟต์แวร์ที่สร้างขึ้นกับ GAMS สามารถเปรียบเทียบผลการคำนวณได้ดังนี้

Software		Optimal solution	Objective value	Iteration		Time
				Cutting	Total	
Cut	Newton	(0.000759 , 1.400000 , 5.279848)	7.079999	10	45	0.11
	Steepest descent	(0.000758 , 1.400000 , 5.279848	7.079999	10	248	1.2
	Combine N_S	(0.000759 , 1.400000 , 5.279848	7.079999	10	55	0.15
GAMS		(0.000 , 1.400 , 5.280)	7.08	51		0.11

ปัญหา NLP5

$$\text{minimize} \quad 2 + x_1^2 + x_2^2 + x_3^2 + x_4^2 + \frac{1}{1 + x_1^2 + x_2^2} + \frac{1}{1 + x_3^2 + x_4^2}$$

$$\begin{aligned} \text{subject to} \quad & x_1 + x_2 + 2x_3 + x_4 \le 6 \\ & -x_1 \le 0 \\ & -x_2 \le 0 \\ & -x_3 \le 0 \\ & -x_4 \le 0 \\ & 2x_1 + x_2 + x_3 + 4x_4 = 7 \end{aligned}$$

ค่าเริ่มต้นที่ได้จากโปรแกรมคือ $\mathbf{x}^{(0)} = (0.808219, 0.876712, 1.821918, 0.671233)^t$

จากการหาผลเฉลยโดยใช้ซอฟต์แวร์ที่สร้างขึ้นกับ GAMS สามารถเปรียบเทียบผลการคำนวณได้ดังนี้

Software		Optimal solution	Objective value	Iteration		Time
				Cutting	Total	
Cut	Newton	(0.761614,0.380807,0.299763,1.199051)	5.227972	2	6	0.03
	Steepest descent	(0.761614,0.380807,0.299763,1.199051)	5.227972	2	120	0.62
	Combine N_S	(0.761614,0.380807,0.299763,1.199051)	5.227972	2	7	0.03
GAMS		(0.762 , 0.0381 , 0.300 , 1.199)	5.228	9		0.05

ปัญหา NLP 6

$$\begin{aligned} \text{minimize} \quad & \ln(x_1^2 + 1) + \ln(x_2^2 + 1) \\ \text{subject to} \quad & 3x_1 + 4x_2 \le 12 \\ & x_1 \ge 1 \\ & x_2 \ge 1 \\ & x_1 - x_2 = 0 \end{aligned}$$

ค่าเริ่มต้นที่ได้จากโปรแกรมคือ $\mathbf{x}^{(0)} = (0.999999, 0.999999)^t$

จากการหาผลเฉลยโดยใช้ซอฟต์แวร์ที่สร้างขึ้นกับ GAMS สามารถเปรียบเทียบผลการคำนวณได้ดังนี้

Software		Optimal solution	Objective value	Iteration		Time
				Cutting	Total	
Cut	Newton					
	Steepest descent	(0.999812 , 0.999812)	1.386241	7	3524	16.54
	Combine N_S					
GAMS		(1.000 , 1.000)	1.386	1		0.05

จากปัญหานี้จะเห็นได้ว่าเราไม่สามารถหาผลเฉลยโดยวิธี Newton และ Combine N_S ที่เป็นเช่นนี้เพราะว่าในขั้นตอนการหาผลเฉลยของทั้ง 2 วิธีมีการหาเมทริกซ์ผกผันของ

$$\begin{bmatrix} -\frac{2(x_1^2 - 1)}{(x_1^2 - 1)^2} & 0 & 1 \\ 0 & -\frac{2(x_2^2 - 1)}{(x_2^2 - 1)^2} & -1 \\ 1 & -1 & 0 \end{bmatrix}$$

เมื่อ (x_1, x_2) เข้าใกล้ (1.000 ,1.000) ซึ่งเป็นผลเฉลยเหมาะที่สุด จะเห็นได้ว่า เมทริกซ์นี้มีดีเทอร์มิแนนต์เท่ากับ 0 ซึ่งทำให้เราไม่สามารถหาเมทริกซ์ผกผันได้

ปัญหา NLP 7

$$\begin{aligned} \text{minimize} \quad & 4e^{\frac{x_1}{4}} + e^{x_2} + 5e^{\frac{x_3}{5}} + e^{x_4} + x_5^2 \\ \text{subject to} \quad & 2x_1 + x_2 + x_3 + 4x_4 - x_5 \le 7 \\ & -x_1 \le 0 \\ & -x_2 \le 0 \\ & -x_3 \le 0 \\ & -x_4 \le 0 \\ & -x_5 \le 0 \\ & x_1 + x_2 + 2x_3 + x_4 + x_5 = 9 \end{aligned}$$

ค่าเริ่มต้นที่ได้จากโปรแกรมคือ $\mathbf{x}^{(0)} = (0.991667, 1.125, 2.383333, 0.725, 1.391667)^t$

จากการหาผลเฉลยโดยใช้ซอฟต์แวร์ที่สร้างขึ้นกับ GAMS สามารถเปรียบเทียบผลการคำนวณได้ดังนี้

Software		Optimal solution	Objective value	Iteration		Time
				Cutting	Total	
Cut	Newton	(0.379685,0.094921,3.940343,0.094921,0.549786)	17.895423	2	7	0.01
	Steepest descent	(0.379699,0.094921,3.940337,0.094921,0.549786)	17.895423	2	515	2.62
	Combine N_S	(0.379684,0.094920,3.940340,0.094921,0.549786)	17.895423	2	29	0.03
GAMS		(0.380 , 0.095 , 3.940 , 0.095 , 0.550)	17.895	12		0.06

ปัญหา NLP 8

$$\text{minimize} \quad x_1^2 + x_2^2 + 2x_3^2 + x_4^2 + 3x_5^2 + 4x_6^2 + 5x_7^2 + x_8^2 + 4x_9^2 + 2x_{10}^2$$

$$\begin{aligned} \text{subject to} \quad & x_1 + x_2 + x_3 \le 10 \\ & x_4 + x_5 + x_6 \le 10 \\ & x_7 + x_8 + x_9 \le 10 \\ & x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9 \ge 0 \\ & 3x_1 + x_2 - x_3 + 3x_4 - 2x_5 - 7x_6 + 4x_7 + 6x_8 - 3x_9 + x_{10} = 3 \end{aligned}$$

ค่าเริ่มต้นที่ได้จากโปรแกรมคือ $\mathbf{x}^{(0)} = (1,0,1,0,1,0,1,0,1,2)^t$

จากการหาผลเฉลยโดยใช้ซอฟต์แวร์ที่สร้างขึ้นกับ GAMS สามารถเปรียบเทียบผลการคำนวณได้ดังนี้

Software		Optimal solution	Objective value	Iteration		Time
				Cutting	Total	
Cut	Newton	(0.153879,0.051293,0.002883,0.153879,-0.001604 0.001668,0.041034,0.307757,-0.000152,0.025646)	0.153284	59	118	0.76
	Steepest descent	(0.154536,0.051512,-0.001135,0.154536,0.000230 0.003471,0.041210,0.309073,0.000045,0.025756)	0.153261	61	5139	38.49
	Combine N_S	(0.153896,0.051346,0.000641,0.153896,-0.001632 0.001552,0.041135,0.307481,-0.000211,0.025713)	0.153275	59	311	1.32
GAMS		(0.153 , 0.051 , 0.000 , 0.153 , 0.000 , 0.000 , 0.041, 0.0307 , 0.000 , 0.026)	0.153	40		0.11

ปัญหา NLP 9

$$\begin{aligned} &\text{mininmize} && (x_1-3)^2+(x_2-2)^2 \\ &\text{subject to} && x_1^2+x_2^2-5 \le 0 \\ &&& -x_1 \le 0 \\ &&& -x_2 \le 0 \\ &&& h(\mathbf{x})=0 \end{aligned}$$

$$h(\mathbf{x})=\begin{cases} x_1+2x_2-4 & ;\mathbf{x}\in\mathbf{D}_1 \\ 4x_1-2x_2-1 & ;\mathbf{x}\in\mathbf{D}_2 \end{cases}$$

เมื่อ $\mathbf{D}_1=\{x_1|-1\le x_1\le 1\}\times\mathbf{R}$ และ $\mathbf{D}_2=\{x_1|1\le x_1\le 2\}\times\mathbf{R}$

จากการหาผลเฉลยโดยใช้ซอฟต์แวร์ที่สร้างขึ้นกับ GAMS สามารถเปรียบเทียบผลการคำนวณได้ดังนี้

			PCS1	PCS2
Initial			(1 , 1.5)	(1 , 1.5)
Newton	Optimal solution		(1.005468 , 1.497266)	(1.194531 , 1.889063)
	Objective value		4.249963	3.270174
	Iteration	Cutting	9	9
		Total	15	17
Steep descent	Optimal solution		(1.005468 , 1.497256)	(1.194524 , 1.889047)
	Objective value		4.249963	3.270174
	Iteration	Cutting	9	9
		Total	266	145
Combine N_S	Optimal solution		(1.005468 , 1.497266)	(1.194530 , 1.889059)
	Objective value		4.249963	3.270174
	Iteration	Cutting	9	9
		Total	24	28
GAMS	Optimal solution		(1.000 , 1.500)	(1.195 , 1.890)
	Objective value		4.25	3.27
	Iteration		1	6

	Newton	Steepest descent	Combine N_S	GAMS
Optimal solution	(1.194531,1.889063)	(1.194524,1.889047)	(1.194530,1.889059)	(1.195 , 1.890)
Global minimum	3.270174	3.270175	3.270174	3.27
Piecewise No.	2	2	2	2
Total time	0.07	1.85	0.09	1.16

ปัญหา NLP10

$$\begin{aligned} \text{mininmize} \quad & \frac{x_1^2}{2}+\frac{x_2^2}{3}+\frac{x_3^2}{4} \\ \text{subject to} \quad & -2x_1+5x_2+x_3-20 \le 0 \\ & x_1, x_2, x_3 \ge 0 \\ & h(\mathbf{x}) = 0 \end{aligned}$$

$$h(\mathbf{x}) = \begin{cases} x_1+4x_2-2x_3+13 & ;\mathbf{x} \in \mathbf{D}_1 \\ x_1-2x_2-x_3+6 & ;\mathbf{x} \in \mathbf{D}_2 \\ 7x_1-4x_2-2x_3+2 & ;\mathbf{x} \in \mathbf{D}_3 \end{cases}$$

เมื่อ $\mathbf{D}_1 = \{x_1 \,|\, -1 \le x_1 \le 1\} \times \mathbf{R}^2, \mathbf{D}_2 = \{x_1 \,|\, 1 \le x_1 \le 2\} \times \mathbf{R}^2$ และ $\mathbf{D}_3 = \{x_1 \,|\, 2 \le x_1 \le 4\} \times \mathbf{R}^2$

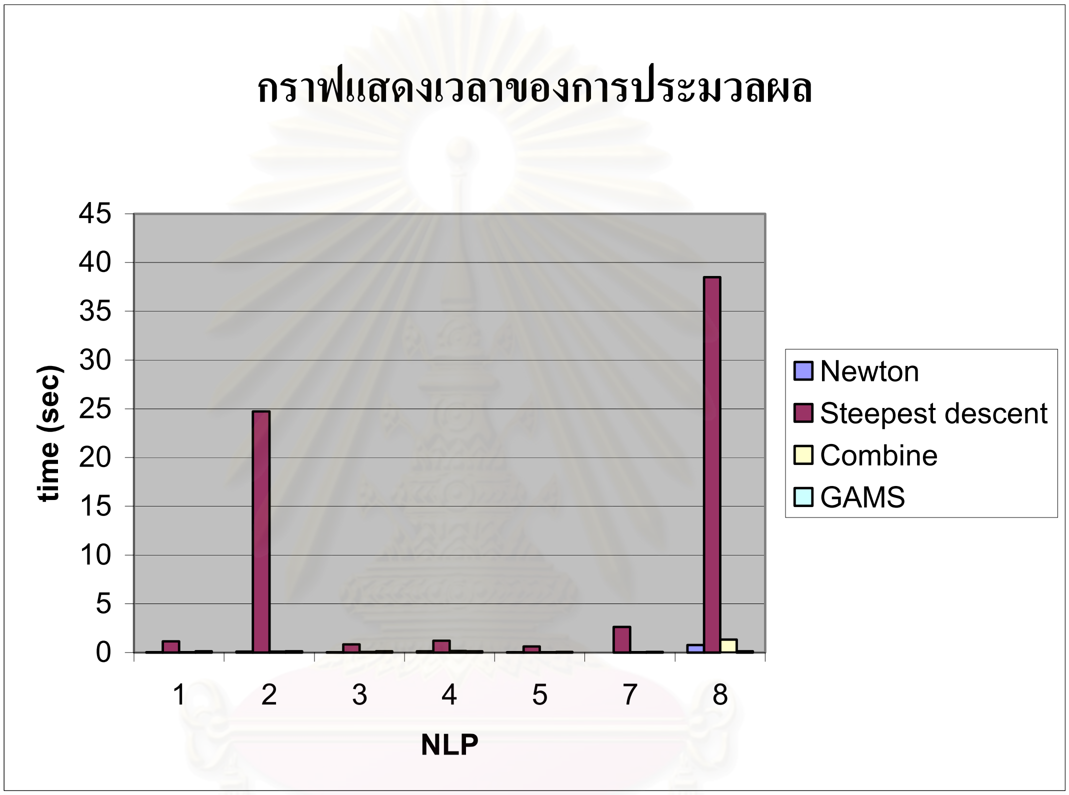
จากการหาผลเฉลยโดยใช้ซอฟต์แวร์ที่สร้างขึ้นกับ GAMS สามารถเปรียบเทียบผลการคำนวณได้ดังนี้

			PCS1	PCS2	PCS3
Initial			(1 , 2 , 3)	(1 , 2 , 3)	(2 , 3 , 2)
Newton	Optimal solution		(-0.0050,2.4366,1.6244)	(0.9913,2.617,1.7478)	(1.9915,2.9889,1.9926)
	Objective value		2.640612	3.562458	5.999909
	Iteration	Cutting	9	10	9
		Total	18	20	18
Steep descent	Optimal solution		(-0.0050,2.4366,1.6244)	(1.0087,2.6283,1.7522)	(1.9915,2.9889,1.9926)
	Objective value		2.640612	3.562457	5.999909
	Iteration	Cutting	9	10	9
		Total	3457	5241	3874
Combine N_S	Optimal solution		(-0.0050,2.4366,1.6244)	(0.9925,2.6191,1.7510)	(1.9917,2.9889,1.9926)
	Objective value		2.640612	3.562459	5.999909
	Iteration	Cutting	9	10	9
		Total	38	55	41
GAMS	Optimal solution		(0.000 , 2.437 , 1.625)	(1.000 , 2.625 , 1.750)	(2.000 , 3.000 , 2.000)
	Objective value		2.641	3.562	6
	Iteration		3	3	3

	Newton	Steepest descent	Combine N_S	GAMS
Optimal solution	(-0.0050,2.4366, 1.6244)	(-0.0050,2.4366, 1.6244)	(-0.0050,2.4366, 1.6244)	(0.000 , 2.437 , 1.625)
Global minimum	2.640612	2.640612	2.640612	2.641
Piecewise No.	1	1	1	1
Total time	0.17	63.3	0.34	0.28

5.2 สรุปผล

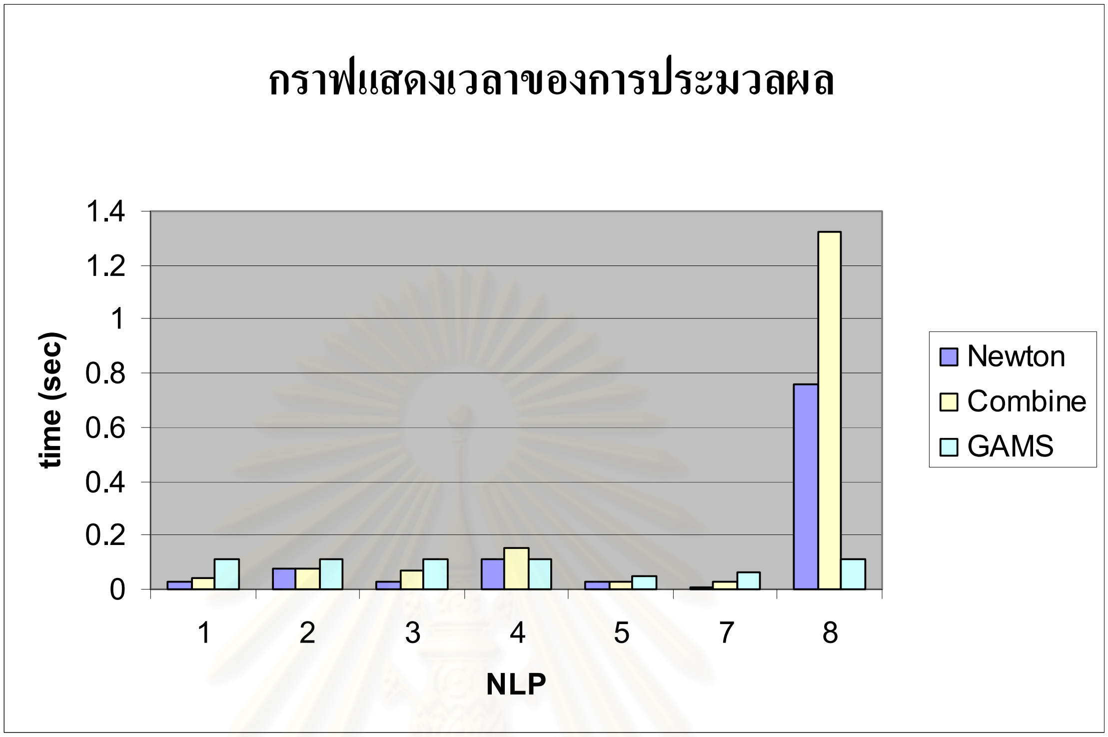
จากการสร้างซอฟต์แวร์เพื่อหาผลเฉลยของปัญหากำหนดการไม่เชิงเส้นภายใต้เงื่อนไขบังคับอสมการไม่เชิงเส้น และสมการเชิงเส้นเป็นช่วงโดยใช้วิธี cutting-plane สามารถหาผลเฉลยได้จริงเมื่อเปรียบเทียบการคำนวณหาผลเฉลยโดยใช้โปรแกรม ผลเฉลยเหมาะที่สุดที่ได้มีค่าเท่ากันหรือแตกต่างกันที่ทศนิยมตำแหน่งที่ 3



รูปที่ 5.1 แสดงถึงเวลาในการประมวลผลของทั้ง 4 วิธีโดยพิจารณาเฉพาะกรณีที่จำนวนช่วงของเงื่อนไขบังคับสมการเชิงเส้นเป็น 1 เท่านั้น

จากรูปที่ 5.1จะเห็นได้ว่าเวลาที่ใช้ในซอฟต์แวร์ที่สร้างขึ้นระหว่างวิธีการหาผลเฉลยระบบสมการไม่เชิงเส้นโดยวิธี Steepest descent กับวิธีอื่น ๆ แตกต่างกันอยู่มาก เวลาที่ใช้ในวิธี Newton และวิธี Combine N_S น้อยกว่าเวลาที่ใช้ในวิธี Steepest descent เนื่องจากการทำซ้ำของวิธี Steepest descentในขั้นตอนการหาผลเฉลยระบบสมการไม่เชิงเส้น มากกว่าทั้ง 2 วิธีนั่นเอง

ดังนั้นเราจะพิจารณาเวลาที่ใช้ในการประมวลผลจากซอฟต์แวร์ที่สร้างขึ้นโดยใช้วิธี Newton และวิธี Combine N_S หาผลเฉลยของระบบสมการไม่เชิงเส้นกับเวลาที่ใช้ในการประมวลผลจากโปรแกรม GAMS ดังแสดงในรูปที่ 5.2



รูปที่ 5.2 แสดงถึงเวลาในการประมวลผลของวิธี Newton, วิธี Combine N_S และเวลาในการประมวลผลจากโปรแกรม GAMS โดยพิจารณาเฉพาะกรณีที่จำนวนช่วงของเงื่อนไขบังคับสมการเชิงเส้นเป็น 1 เท่านั้น

จากรูปที่ 5.2 จะเห็นได้ว่าเวลาที่ใช้ในการประมวลผลจากซอฟต์แวร์ที่สร้างขึ้นโดยใช้วิธี Combine N_S จะใช้เวลามากกว่าวิธี Newton เนื่องจากมีจำนวนการทำซ้ำในขั้นตอนการหาผลเฉลยของระบบสมการไม่เชิงเส้นมากกว่าเพราะมีการใช้วิธี Steepest descent ในช่วงแรกจากนั้นจึงเปลี่ยนไปใช้วี Newton

สำหรับเวลาที่ใช้ในการประมวลผลจากซอฟต์แวร์ที่สร้างขึ้นโดยใช้วิธี Newton น้อยกว่าเวลาที่ใช้ในการประมวลผลจากโปรแกรม GAMS เมื่อปัญหามีจำนวนตัวแปรอิสระน้อย หรือค่าเริ่มต้น $\mathbf{x}^{(0)}$ อยู่ใกล้กับผลเฉลยจริง

ข้อเสนอแนะเกี่ยวกับงานวิจัยขั้นต่อไป ศึกษาวิธีการพัฒนาซอฟต์แวร์นี้ให้ผู้ใช้สามารถใช้งานได้ง่ายขึ้น ไม่จำเป็นต้องแก้ไขไฟล์ทุกครั้งที่ต้องการใช้งานจริง และหาวิธีการที่จะขยายขอบเขตของปัญหาให้ใช้ได้ในกรณีที่ฟังก์ชันเป็นฟังก์ชันนูนแต่ไม่สามารถหาอนุพันธ์ได้เช่น ฟังก์ชันค่าสัมบูรณ์

รายการอ้างอิง

[1] Kaczor, W. J., Nowak, M. T., Problem in Mathematical Analysis I Real Numbers: Sequence and Series, English, A.M.S., 2000.

[2] Peressini, A. L., Sullivan, F. E., and Uhl, J. J., The Mathematics of Nonlinear Programming. NewYork: Springer-Verlag, 1988.

[3] Bazaraa, M. S., Sherali, H. D., and Shetty, C. M., Nonlinear Programming Theory and Algorithm. New York: John Wiley & Sons, 1993.

[4] Hillier, F., Lieberman, J. G., Introduction to Operations Research. New York: McGraw-Hill, 1995.

[5] Burden, R. L., Faires, J. D., Numerical Analysis. California: Brooks/Cole Publishing Company, 1997.

[6] Griewank, A., Juedes, D., Mitey, H., Utke, J., Vogel, O., and Walther, A., ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++. Dresden: Institute of Science Computing, Technical University Dresden, Germany, 1996. (Unpublished Manuscript)

[7] Makhorin, A., GNU Linear Programming Kit. Moscow: Department for Applied Informatics, Moscow Aviation Institute, Russia, 2002. (Unpublished Manuscript)

ภาคผนวก

ภาคผนวก ก

Solution of system of nonlinear equation

ขั้นตอนในการหาผลเฉลยของระบบสมการไม่เชิงเส้นนั้นยุ่งยากกว่าการผลเฉลยของระบบสมการเชิงเส้นอยู่มาก โดยส่วนมากเราจะหลีกเลี่ยงฟังก์ชันไม่เชิงเส้น โดยการหาฟังก์ชันเชิงเส้นมาประมาณฟังก์ชันไม่เชิงเส้น แต่ถ้าเมื่อใดที่การหาฟังก์ชันเชิงเส้นที่ใช้ประมาณฟังก์ชันไม่เชิงเส้นไม่สามารถทำได้ หรือเป็นไปได้ยากแล้ว เราจะใช้วิธีการประมาณค่าผลเฉลยของสมการไม่เชิงเส้นมาประยุกต์ใช้แทน

Newton method

วิธี Newton สำหรับการประมาณค่าผลเฉลย x สำหรับปัญหาสมการไม่เชิงเส้น

$$f(x)=0$$

เมื่อ $f:\mathbf{R}\to\mathbf{R}$ ต้องการ ค่าเริ่มต้น $x^{(0)}\in\mathbf{R}$ และนิยามลำดับของ $x^{(k)}$ โดย

$$x^{(k)}=x^{(k-1)}-\frac{f(x^{(k-1)})}{f'(x^{(k-1)})}$$

สำหรับการหาผลเฉลย $\mathbf{x}$ สำหรับปัญหา

$$\mathbf{F}(\mathbf{x})=\mathbf{0}$$

โดยที่ $\mathbf{F}(\mathbf{x})=\begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}$ และ $f_1:\mathbf{R}^{\mathrm{n}}\to\mathbf{R}, f_2:\mathbf{R}^{\mathrm{n}}\to\mathbf{R},\ldots, f_m:\mathbf{R}^{\mathrm{n}}\to\mathbf{R}$

เราต้องการ ค่าเริ่มต้น $\mathbf{x}^{(0)}\in\mathbf{R}^n$ และนิยามลำดับของ $\mathbf{x}^{(k-1)}$ โดย

$$\mathbf{x}^{(k)}=\mathbf{x}^{(k-1)}-\left[\mathbf{J}(\mathbf{x}^{(k-1)})\right]^{-1}\mathbf{F}(\mathbf{x}^{(k-1)}) \qquad : k\ \geq 1$$

เมื่อ $\mathbf{J}(\mathbf{x})$ คือเมทริกซ์ Jacobian

$$\mathbf{J}(\mathbf{x})=\begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \frac{\partial f_m(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{bmatrix}$$

ขนาดของเมทริกซ์ $\mathbf{J}(\mathbf{x})$ เกิดขึ้นได้ 3 กรณีคือ $m>n$, $m<n$ และ $m=n$ เมื่อ m เป็นจำนวนสมการไม่เชิงเส้น และ n เป็นจำนวนตัวแปรในระบบสมการ ในกรณีที่ $m>n$ และ $m<n$

เราไม่สามารถหาดีเทอร์มิแนนต์ของ $\mathbf{J}(\mathbf{x})$ ได้ ดังนั้นการหา $\left[\mathbf{J}(\mathbf{x}^{(k-1)})\right]^{-1}$ ต้องใช้ QR-factorization ระบบสมการไม่เชิงเส้นในวิทยานิพนธ์นี้เราสนใจเพียงกรณีเดียวคือ $m = n$ เนื่องจาก

$$S(x_1, x_2, \ldots x_n, w) = f(x_1, x_2, \ldots x_n) + \mathbf{u}_k^t \mathbf{g}(x_1, x_2, \ldots x_n) + wh(x_1, x_2, \ldots x_n)$$

เราต้องการหาผลเฉลย $(x_1, x_2, \ldots x_n)$ ที่ทำให้ $\nabla S(x_1, x_2, \ldots x_n, w) = \mathbf{0}$ นั่นคือ

$$\begin{aligned}
\frac{\partial S(x_1, x_2, \ldots x_n, w)}{\partial x_1} &= 0\\
\frac{\partial S(x_1, x_2, \ldots x_n, w)}{\partial x_2} &= 0\\
&\vdots\\
\frac{\partial S(x_1, x_2, \ldots x_n, w)}{\partial x_n} &= 0\\
\frac{\partial S(x_1, x_2, \ldots x_n, w)}{\partial w} &= 0
\end{aligned}$$

จะเห็นได้ว่าระบบสมการนี้มีจำนวนตัวแปรเท่ากับจำนวนสมการ

เราจำเป็นต้องหา เมทริกซ์ $\left[\mathbf{J}(\mathbf{x}^{(k-1)})\right]^{-1}$ ทุกการทำซ้ำ ในทางปฏิบัติเราจึงหลีกเลี่ยงโดยการหา $\mathbf{y}^{(k-1)}$ ที่สอดคล้อง

$$\mathbf{J}(\mathbf{x}^{(k-1)})\mathbf{y}^{(k-1)} = -\mathbf{F}(\mathbf{x}^{(k-1)})$$

นั่นคือ $$\mathbf{y}^{(k-1)} = -\left[\mathbf{J}(\mathbf{x}^{(k-1)})\right]^{-1}\mathbf{F}(\mathbf{x}^{(k-1)})$$

ดังนั้น $$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \mathbf{y}^{(k-1)} \quad : k \geq 1$$

<u>ตัวอย่าง</u> ระบบสมการไม่เชิงเส้น

$$3x_1 - \cos(x_2 x_3) - \frac{1}{2} = 0,$$

$$x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0,$$

$$e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0$$

เมื่อกำหนดให้ $\mathbf{x}^{(0)} = (0.1, 0.1, -0.1)^t$ แล้ววิธี Newton จะลู่เข้าสู่คำตอบและมีผลเฉลยโดยประมาณอยู่ที่ $(0.5, 0, -0.52359877)^t$

เมทริกซ์ Jacobian สำหรับระบบสมการไม่เชิงเส้นนี้คือ

$$\mathbf{J}(x_1, x_2, x_3) = \begin{bmatrix} 3 & x_3 \sin x_2 x_3 & x_2 \sin x_2 x_3 \\ 2x_1 & -162(x_2 + 0.1) & \cos x_3 \\ -x_2 e^{-x_1 x_2} & -x_1 e^{-x_1 x_2} & 20 \end{bmatrix}$$

และ

$$\begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{bmatrix} = \begin{bmatrix} x_1^{(k-1)} \\ x_2^{(k-1)} \\ x_3^{(k-1)} \end{bmatrix} + \begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix}$$

เมื่อ

$$\begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix} = -\left(\mathbf{J}(x_1^{(k-1)}, x_2^{(k-1)}, x_3^{(k-1)})\right)^{-1} \mathbf{F}(x_1^{(k-1)}, x_2^{(k-1)}, x_3^{(k-1)})$$

ดังนั้นที่การทำซ้ำที่ k เราต้องการผลเฉลยของระบบสมการเชิงเส้น

$$\mathbf{J}(\mathbf{x}^{(k-1)})\mathbf{y}^{(k-1)} = -\mathbf{F}(\mathbf{x}^{(k-1)})$$

เมื่อ

$$\mathbf{J}\left(\mathbf{x}^{(k-1)}\right) = \begin{bmatrix} 3 & x_3^{(k-1)} \sin x_2^{(k-1)} x_3^{(k-1)} & x_2^{(k-1)} \sin x_2^{(k-1)} x_3^{(k-1)} \\ 2x_1^{(k-1)} & -162\left(x_2^{(k-1)} + 0.1\right) & \cos x_3^{(k-1)} \\ -x_2^{(k-1)} e^{-x_1^{(k-1)} x_2^{(k-1)}} & -x_1^{(k-1)} e^{-x_1^{(k-1)} x_2^{(k-1)}} & 20 \end{bmatrix}$$

เราอาจใช้วิธี Finite difference ในการประมาณค่าอนุพันธ์ หรือเรียกใช้โปรแกรม ADOL-C เพื่อคำนวณหาอนุพันธ์ที่ $\mathbf{x}^{(k-1)}$ ได้

ผลลัพธ์ที่ได้จากการทำซ้ำมีดังนี้

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\left\lVert \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right\rVert$
0	0.10000000	0.10000000	-0.10000000	
1	0.50003712	0.01946686	-0.52152047	0.422
2	0.00158859	0.00158859	-0.52355711	0.00179
3	0.50000034	0.00001244	-0.52359845	0.000158
4	0.50000000	0.00000000	-0.52359877	0.00000124
5	0.50000000	0.00000000	-0.52359877	0

จากตัวอย่างที่ผ่านมา วิธี Newton สามารถลู่เข้าสู่ผลเฉลยได้เร็วมาก อย่างไรก็ตามวิธีนี้ต้องการค่าเริ่มต้น $\mathbf{x}^{(0)}$ ที่ดีพอ ไม่เช่นนั้นแล้ว วิธี Newton นี้อาจจะไม่ลู่เข้าสู่ผลเฉลยเช่น เมื่อเราเลือก $\mathbf{x}^{(0)} = (100, 0, 100)^t$ จะได้ผลลัพธ์จากการทำซ้ำดังนี้

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\left\Vert \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right\Vert$
0	100.00000000	0.00000000	100.00000000	
1	0.50000000	-840.05711662	4200.80918189	4.300809e+03
2	0.74877177	-420.09211710	-6300.89591726	2.100087e+03
3	1.12072258	-210.07671804	-9450.87995117	3.149984e+03
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
9	-2594.75322379	30.54168612	-0.47359878	1.489677e+05
10	nan	nan	nan	

Steepest descent

ข้อดีของวิธี Newton ที่ใช้สำหรับหาผลเฉลยของระบบสมการไม่เชิงเส้นคือ ลู่เข้าสู่ผลเฉลยได้เร็ว แต่เราจำเป็นต้องรู้ค่าเริ่มต้น $\mathbf{x}^{(0)}$ ที่ใกล้กับผลเฉลยมากพอ วิธี steepest descent ที่จะกล่าวถึงต่อไปนี้ เป็นวิธีที่ไม่สนใจค่าเริ่มต้น $\mathbf{x}^{(0)}$ ว่าจะมีค่าเท่าใด แต่ใช้เวลาในการลู่เข้าสู่ผลเฉลยช้ามาก

ระบบสมการไม่เชิงเส้น

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, x_2, \ldots, x_n) &= 0 \end{aligned}$$

จะมีผลเฉลยอยู่ที่ $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ เมื่อฟังก์ชัน g นิยามโดย

$$g(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{m} [f_i(x_1, x_2, \ldots, x_n)]^2$$

มีค่าต่ำสุดเท่ากับ 0

จากทฤษฎี Extreme value ฟังก์ชันตัวแปรเดียวที่สามารถหาอนุพันธ์ได้จะมีค่าต่ำสุดสัมพัทธ์ เมื่ออนุพันธ์ของฟังก์ชันมีค่าเท่ากับ 0 เราสามารถขยายผลนี้มาใช้ในกรณีที่เป็นฟังก์ชันหลายตัวแปร โดยอาศัย gradient ของฟังก์ชัน เมื่อ

$$\nabla g(\mathbf{x}) = \left(\frac{\partial g(\mathbf{x})}{\partial x_1}, \frac{\partial g(\mathbf{x})}{\partial x_2}, \cdots, \frac{\partial g(\mathbf{x})}{\partial x_n} \right)$$

กล่าวคือฟังก์ชัน g จะมีค่าต่ำสุดสัมพัทธ์ที่ $\mathbf{x}$ ถ้า $\nabla g(\mathbf{x}) = \mathbf{0}$ หลักการของวิธี steepest descent คือ การหาค่า $\mathbf{x}$ ที่ทำให้ $\nabla g(\mathbf{x}) = \mathbf{0}$ และนิยามลำดับของ $\mathbf{x}^{(k)}$ โดย

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \alpha \nabla g\left(\mathbf{x}^{(k-1)}\right) \qquad : k \geq 1$$

เมื่อ $-\nabla g\left(\mathbf{x}^{(k-1)}\right)$ คือทิศทางที่ทำให้ $g\left(\mathbf{x}^{(k-1)}\right)$ มีค่าลดลงมากที่สุด และ $g\left(\mathbf{x}^{(k)}\right) < g\left(\mathbf{x}^{(k-1)}\right)$ สำหรับการเลือกค่า α เราจะพิจารณาฟังก์ชันของหนึ่งตัวแปร $h(\alpha) = g\left(\mathbf{x}^{(k-1)} - \alpha \nabla g\left(\mathbf{x}^{(k-1)}\right)\right)$ เราจะใช้ α ที่ทำให้ค่า $h(\alpha)$ มีค่าน้อยที่สุด นั่นคือการหาผลเฉลยของ

$$\begin{array}{ll} \text{minimize} & h(\alpha) = g\left(\mathbf{x}^{(k-1)} - \alpha \nabla g\left(\mathbf{x}^{(k-1)}\right)\right) \\ \text{subject to} & \alpha > 0 \end{array}$$

การหาค่าที่ต่ำที่สุดของ $h(\alpha)$ โดยวิธีตรง จำเป็นต้องใช้อนุพันธ์ของ $h(\alpha)$ และหาผลเฉลยของสมการไม่เชิงเส้นตัวแปรเดียวซึ่งใช้เวลานาน Burden, R. L. และ Faires, J. D. [5] ได้นำเสนอวิธีการประมาณ $h(\alpha)$ ด้วยพหุนามกำลังสอง $P(\alpha)$ และจำนวนจริง 3 จำนวน $\alpha_1, \alpha_2, \alpha_3$ ให้เข้าใกล้กับค่าที่ต่ำที่สุดของ $h(\alpha)$ มากที่สุด โดยใช้ Newton's Interpolatory Divided-Difference Formula

$$\begin{aligned} \mathrm{P_n}(a) = {} & f[a_0] + f[a_0, a_1](a - a_0) + f[a_0, a_1, a_2](a - a_0)(a - a_1) \\ & + \ldots + f[a_0, a_1, a_2, \ldots, a_n](a - a_0)(a - a_1)\ldots(a - a_{n-1}) \end{aligned}$$

เมื่อ $n = 2$

$$\mathrm{P_2}(a) = f[a_0] + f[a_0, a_1](a - a_0) + f[a_0, a_1, a_2](a - a_0)(a - a_1)$$

เมื่อ $f[a_i] = f(a_i)$

$$f[a_i, a_{i+1}] = \frac{f[a_{i+1}] - f[a_i]}{a_{i+1} - a_i}$$

$$f[a_i, a_{i+1}, a_{i+2}] = \frac{f[a_{i+1}, a_{i+2}] - f[a_i, a_{i+1}]}{a_{i+2} - a_i}$$

ดังแสดงในตัวอย่าง

<u>**ตัวอย่าง**</u> ระบบสมการไม่เชิงเส้น

$$\begin{aligned} 3x_1 - cos(x_2 x_3) - \frac{1}{2} &= 0, \\ x_1^2 - 81(x_2 + 0.1)^2 + sin\, x_3 + 1.06 &= 0, \\ e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} &= 0 \end{aligned}$$

หาผลเฉลยของระบบสมการไม่เชิงเส้น โดยวิธี steepest descent

ให้ $\mathbf{x}^{(0)} = (0, 0, 0)^t$

และ

$$g(x_1,x_2,x_3)=[f_1(x_1,x_2,x_3)]^2+[f_2(x_1,x_2,x_3)]^2+[f_3(x_1,x_2,x_3)]^2$$

เมื่อ

$$\begin{aligned}\nabla g(x_1,x_2,x_3)\equiv\nabla g(\mathbf{x}) &= \Bigg(2f_1(\mathbf{x})\frac{\partial f_1(\mathbf{x})}{\partial x_1}+2f_2(\mathbf{x})\frac{\partial f_2(\mathbf{x})}{\partial x_1}+2f_3(\mathbf{x})\frac{\partial f_3(\mathbf{x})}{\partial x_1},\\ &\qquad 2f_1(\mathbf{x})\frac{\partial f_1(\mathbf{x})}{\partial x_2}+2f_2(\mathbf{x})\frac{\partial f_2(\mathbf{x})}{\partial x_2}+2f_3(\mathbf{x})\frac{\partial f_3(\mathbf{x})}{\partial x_2},\\ &\qquad 2f_1(\mathbf{x})\frac{\partial f_1(\mathbf{x})}{\partial x_3}+2f_2(\mathbf{x})\frac{\partial f_2(\mathbf{x})}{\partial x_3}+2f_3(\mathbf{x})\frac{\partial f_3(\mathbf{x})}{\partial x_3}\Bigg)\\ &= 2\mathbf{J}(\mathbf{x})^t\mathbf{F}(\mathbf{x})\end{aligned}$$

สำหรับ $\mathbf{x}^{(0)}=(0,0,0)^t$

จะได้

$$g(\mathbf{x}^{(0)})=111.975, \qquad \nabla g(\mathbf{x}^{(0)})=(-0.0214514,-0.0193062,0.999583)^t$$

เนื่องจาก $\mathbf{x}^{(1)}=\mathbf{x}^{(0)}-\alpha\nabla g(\mathbf{x}^{(0)})$ ดังนั้นเราจะต้องหา α ที่ทำให้ $g(\mathbf{x}^{(1)})$ มีค่าน้อยที่สุดโดย Newton's Interpolatory Divided-Difference Formula

ให้ $k=1,\ a_0=\alpha_1=0, a_1=\alpha_2=0.5, a_2=\alpha_3=1$ และ $G_i=g\left(\mathbf{x}^{(k-1)}-\alpha_i\nabla g\left(\mathbf{x}^{(k-1)}\right)\right)$

$$f[a_0]=G_1$$

$$f[a_0,a_1]=\frac{G_2-G_1}{\alpha_2-\alpha_1}=H_1$$

$$f[a_1,a_2]=\frac{G_3-G_2}{\alpha_3-\alpha_2}=H_2$$

$$f[a_0,a_1,a_2]=\frac{f[a_1,a_2]-f[a_0,a_1]}{\alpha_3-\alpha_1}=\frac{H_2-H_1}{\alpha_3-\alpha_1}=H_3$$

$$P(\alpha)=G_1+H_1\alpha+H_3\alpha(\alpha-\alpha_2)$$

ดังนั้น

$\alpha_1=0,\quad G_1=111.975,$

$\alpha_2=0.5,\quad G_2=2.53357,\quad H_1=-218.878,$

$\alpha_3=1,\quad G_3=93.5649,\quad H_2=182.059,\quad H_3=400.937$

และ

$$P(\alpha)=111.975-218.878\alpha+400.937\alpha(\alpha-0.5)$$

เนื่องจาก $P(\alpha)$ มีค่าน้อยสุดที่ $P(\alpha_0)=2.32424$ ดังนั้น $\alpha_0=0.522959$ และ

$$\mathbf{x}^{(1)}=\mathbf{x}^{(0)}-0.522959\left[\nabla g(\mathbf{x}^{(0)})\right]=(0.0112182,0.0100964,-0.522741)^t$$

ดังนั้น $g\left(\mathbf{x}^{(1)}\right)=2.32762$

ผลลัพธ์ที่ได้จากการทำซ้ำมีดังนี้

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$g\left(x_1^{(k)},x_2^{(k)},x_3^{(k)}\right)$
0	0.0	0.0	0.0	111.975
1	0.0112182	0.0100964	-0.522741	2.32762
2	0.137860	-0.205453	-0.522059	1.27406
3	0.266959	0.00551102	-0.558494	1.06813
4	0.272734	-0.00811751	-0.522006	0.468309
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
119	0.49898101	-0.00006058	-0.52362759	0

จากตารางข้างต้น จะเห็นได้ว่า วิธี steepest descent มีการลู่เข้าสู่ผลเฉลยช้ามาก หลังจากการทำซ้ำครั้งที่ 1 ค่า $g\left(x_1^{(k)},x_2^{(k)},x_3^{(k)}\right)$ มีค่าลดลงน้อยมาก เมื่อเทียบกับการทำซ้ำครั้งแรก แต่ข้อดีของวิธีนี้คือ เราสามารถใช้ค่าเริ่มต้น $\mathbf{x}^{(0)}$ ใดก็ได้ จากตัวอย่างที่ผ่านมาเมื่อกำหนดให้ $\mathbf{x}^{(0)}=(100,0,100)^t$ จะเห็นได้ว่าวิธี Newton ไม่ลู่เข้าสู่ผลเฉลย สำหรับวิธี steepest descent ผลลัพธ์ที่ได้จากการทำซ้ำมีดังนี้

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$g\left(x_1^{(k)},x_2^{(k)},x_3^{(k)}\right)$
0	100	0	100	100278416.54681215
1	99.01634906	0.17847847	99.97599311	100134057.04954873
2	98.04437641	0.41231675	99.95172866	96126418.29077032
3	97.12553776	0.80624910	99.92821595	91863600.73902583
4	96.33327751	1.41607970	99.90747162	86822428.75550768
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
50643	0.50102151	0.00006087	-0.52357041	0.00000968

Combination of Newton and steepest descent (combine N_S)

วิธี Newton มีการลู่เข้าสู่ผลเฉลยเร็วมาก แต่ต้องการค่าเริ่มต้น $\mathbf{x}^{(0)}$ ที่ใกล้เคียงผลเฉลยมากพอ แต่วิธี steepest descent สามารถใช้ค่าเริ่มต้นใดก็ได้ แต่มีการลู่เข้าสู่ผลเฉลยช้ามาก เมื่อนำข้อดีของแต่ละวิธีมารวมกัน โดยเริ่มจากการหาผลเฉลยโดยวิธี steepest จนกระทั่ง

$$\left|g^{(k)}(x_1,x_2,x_3)-g^{(k+1)}(x_1,x_2,x_3)\right|<\varepsilon$$

จากนั้นจึงเปลี่ยนไปใช้วิธี Newton หาผลเฉลยต่อไป โดยใช้ $\mathbf{x}^{(k+1)}$ เป็นค่าเริ่มต้นของวิธี Newton โดยที่ $g^{(k)}(x_1,x_2,x_3)=\left[f_1\left(x_1^k,x_2^k,x_3^k\right)\right]^2+\left[f_2\left(x_1^k,x_2^k,x_3^k\right)\right]^2+\left[f_3\left(x_1^k,x_2^k,x_3^k\right)\right]^2$ สำหรับในวิทยานิพนธ์นี้ เราเลือก $\varepsilon=1$ เพราะว่าถ้าเลือก ε ที่มีค่าน้อยมากจะทำให้เสียเวลาการทำซ้ำในขั้นตอน steepest descent มากไป แต่ถ้าเลือก ε ที่มีค่ามากเกินไป อาจทำให้วิธี Newton ไม่ลู่เข้าสู่ผลเฉลย

__ตัวอย่าง__ ระบบสมการไม่เชิงเส้น

$$\begin{aligned}3x_1-cos(x_2x_3)-\frac{1}{2}&=0,\\ x_1^2-81(x_2+0.1)^2+sin\,x_3+1.06&=0,\\ e^{-x_1x_2}+20x_3+\frac{10\pi-3}{3}&=0\end{aligned}$$

หาผลเฉลยของระบบสมการไม่เชิงเส้น โดยวิธี combine N_S ให้ $\mathbf{x}^{(0)}=(100,0,100)^t$ ผลลัพธ์ที่ได้จากการทำซ้ำมีดังนี้

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$g\left(x_1^{(k)},x_2^{(k)},x_3^{(k)}\right)$
0	100	0	100	100278416.54681215
1	99.01634906	0.17847847	99.97599311	100134057.04954873
2	98.04437641	0.41231675	99.95172866	96126418.29077032
3	97.12553776	0.80624910	99.92821595	91863600.73902583
⋮	⋮	⋮	⋮	⋮
20892	65.43936752	7.17340556	2.48467886	41643.82332295
20893	65.43892838	7.17250815	2.48469276	41642.82324227

จะเห็นได้ว่า $\left|g^{(20892)}(x_1,x_2,x_3)-g^{(20893)}(x_1,x_2,x_3)\right|<1$ ดังนั้นจึงเปลี่ยนไปใช้วิธี Newton หาผลเฉลย โดยที่ $\mathbf{x}^{(0)}$ ของวิธี Newton คือ (65.43892838 , 7.17250815 , 2.48469276)

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\left\Vert \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right\Vert$
20894	11.80179229	1.40611448	0.47359878	7.724072e+01
20895	-0.56251311	-0.03489793	-0.47359888	1.236431e+01
20896	0.50005938	-0.00896540	-0.52322072	6.245373e-02
20897	0.50000336	0.00036949	-0.52358910	8.595916e-03
20898	0.50000001	0.00000068	-0.52359876	3.688071e-04
20899	0.50000000	0.00000000	-0.52359878	6.814229e-07

ภาคผนวก ข

โปรแกรมคำนวณหาค่าอนุพันธ์ของฟังก์ชัน ADOL-C

Automatic differentiable of algorithms written in C/C++ หรือ ADOL-C คือซอฟต์แวร์ที่ใช้คำนวณหาค่าอนุพันธ์ของฟังก์ชันโดยใช้วิธีการ Automatic differentiable พัฒนาขึ้นโดย Griewank, A., Juedes, D., Mitey, H., Utke, J., Vogel, O., และ Walther, A. นักวิทยาศาสตร์ของสถาบัน Institute of Scientific Computing Technical University Dresden ประเทศเยอรมนี ในปี ค.ศ. 1999 เขียนโดยใช้ภาษา C/C++ มีการสร้าง header file และ library file ขึ้นมาใช้เอง โดยทั่วไปหลักไวยากรณ์ในการเขียนโปรแกรมใน ADOL-C จะเหมือนกับภาษา C และ C++ จะมีแตกต่างกันอยู่บ้าง เช่น ชนิดของตัวแปร และฟังก์ชันการใช้งานเฉพาะที่มีอยู่ในโปรแกรม ADOL-C เท่านั้น

<u>ตัวอย่าง</u> การใช้งานโปรแกรม ADOL-C เพื่อหา Gradient และ Hessian ของ

$$f(x_1,x_2,x_3) = (x_1-3)^2 - (x_2-2)^2 + x_3(x_1+2x_2-4)$$

ที่จุด $\mathbf{x}^{(0)} = (0,0,0)$

```
#include "adolc.h"                // ประกาศ header file ของ ADOL-C
#include <iostream.h>             // ประกาศ header file ของ C

int main(void)
{
        int I,J,n;
        int tag=1;                // กำหนดหมายเลขให้กับเทป
        n=3;                      // กำหนด จำนวนตัวแปรอิสระ
        double xp[n];
        double yp;
        double Grad[n];
        double Hess[n][n];
        adouble *x = new adouble[n];
```

```
        adouble y;
        for(I=0 ;I<n; I++)
           xp[I]=0;                              // กำหนด x เริ่มต้น
        trace(on(tag);                           // เริ่มเทป
        for(I=0 ;I<n; I++)
           x[I] <<= xp[I];
        y=(x[0]-3)*( x[0]-3)- (x[1]-2)*( x[1]-2)+x[2]*(x[0]+2*x[1]-4)      // กำหนดฟังก์ชัน
        y >>= yp;                                // กำหนดค่าให้ตัวแปรตาม
        delete[] x;                              // ลบตัวแปร
        trace_off(tag);                          // ปิดเทป
        gradient(tag,n,xp,Grad);            // ฟังก์ชันหาค่า Gradient
        printf(" Gradient of f(x,y)\n ");
        for(I=0 ;I<n; I++)
                printf(" %.3lf\t ",Grad[I]);
        hessian(tag,n,xp,Hess);             // ฟังก์ชันหาค่า Hessian
        printf(" Hessian of f(x,y)\n ");
        for(I=0 ;I<n; I++)
          { for(J=0 ;J<n; J++)
                printf(" %.3lf\t " Hess[I][J]);
            printf(" \n ");
          }
        return 0;
}
```

หลังจากได้ชุดคำสั่งหลักแล้ว ขั้นตอนต่อไปคือสั่งให้ตัวแปรภาษาซีทำการแปลชุดคำสั่งด้วยคำสั่งต่อไปนี้

```
$g++ -c –I../SRC adol.c –o adol.out
```

คำสั่งนี้หมายถึง ให้แปลชุดคำสั่งที่อยู่ในไฟล์ที่ชื่อ adol.c โดยใช้ header file ที่อยู่ในแฟ้มข้อมูลที่ชื่อ SRC แล้วให้เก็บผลลัพธ์ที่ได้ไว้ในไฟล์ที่ชื่อ adol.out ซึ่งเป็นไฟล์ที่สามารถใช้งานได้และคำสั่งต่อไปนี้เป็นคำสั่งในการเรียกใช้งานไฟล์ adol.out

```
$./adol.out
```

และจะได้ผลลัพธ์คือ gradient และ Hessian ของ $f(x_1, x_2, x_3)$ ที่จุด $\mathbf{x}^{(0)} = (0,0,0)$

```
gradient of f(x,y)
-6.000          4.000          -4.000
Hessian of f(x,y)
2       0       0
0       -2      0
1       2       0
```

จากตัวอย่างจะเห็นได้ว่า คำสั่ง Hessian จะคำนวณเพียงเมทริกซ์สามเหลี่ยมล่างของ $\nabla^2 f(\mathbf{x}^{(0)})$ เท่านั้น ค่าในตำแหน่งอื่นจะเป็น 0 ทั้งหมด และอนุพันธ์ที่หาจาก ADOL-C จะได้เป็นจำนวนจริง

โปรแกรมคำนวณหาผลเฉลยกำหนดการเชิงเส้น GLPK

GNU Linear Programming Kit หรือ GLPK เป็นโปรแกรมสำหรับหาผลเฉลยของปัญหากำหนดการเชิงเส้นโดยวิธี Revised Simplex method พัฒนาขึ้นโดยนักวิทยาการคอมพิวเตอร์ชาวรัสเซีย Makhorin, A. [6], Department for Applied Informatics, Moscow Aviation Institute ในปี ค.ศ. 2000 มีการสร้าง header file และ library file ขึ้นมาใช้เอง

<u>ตัวอย่าง</u> การใช้งานโปรแกรม GLPK เพื่อหาผลเฉลยของปัญหากำหนดการเชิงเส้น

$$
\begin{array}{llllllll}
\text{maximize} & z = 10x_1 + 6x_2 + 4x_3 & & & & & & \\
\text{subject to} & x_1 & + & x_2 & + & x_3 & \leq & 100 \\
 & 10x_1 & + & 4x_2 & + & 5x_3 & \leq & 600 \\
 & 2x_1 & + & 2x_2 & + & 6x_3 & \leq & 300 \\
 & & & & & x_1, x_3, x_3 & \geq & 0
\end{array}
$$

จากปัญหาดังกล่าว กำหนดให้

$$
\begin{aligned}
p &= x_1 + x_2 + x_3 \\
q &= 10x_1 + 4x_2 + 5x_3 \\
r &= 2x_1 + 2x_2 + 6x_3
\end{aligned}
$$

เมื่อ p, q และ r คือตัวแปรที่กำหนดขึ้นมาเพื่อความสะดวกในการอ้างอิง ดังนั้นจะได้ว่า

$$
\begin{array}{ccccc}
-\infty & < & p & \leq & 100 \\
-\infty & < & q & \leq & 600 \\
-\infty & < & r & \leq & 300
\end{array}
$$

และ

$$
\begin{array}{ccccc}
0 & \leq & x_1 & < & +\infty \\
0 & \leq & x_2 & < & +\infty \\
0 & \leq & x_3 & < & +\infty
\end{array}
$$

จากข้อมูลของปัญหากำหนดการเชิงเส้นที่ได้ เมื่อนำไปสร้างเป็นชุดคำสั่งภาษาซี พร้อมกับการเรียกใช้ชุดคำสั่งของ GLPK

```
/* sample.c */
#include <stdio.h>
#include <stdlib.h>
#include "glpk.h"
```

```
int main(void)
{    LPX *lp;
     int rn[1+9], cn[1+9];
     double a[1+9], Z, x1, x2, x3;

s1:  lp = lpx_create_prob();
s2:  lpx_set_prob_name(lp, "sample");

s3:  lpx_add_rows(lp, 3);

s4:  lpx_set_row_name(lp, 1, "p");
s5:  lpx_set_row_bnds(lp, 1, LPX_UP, 0.0, 100.0);
s6:  lpx_set_row_name(lp, 2, "q");
s7:  lpx_set_row_bnds(lp, 2, LPX_UP, 0.0, 600.0);
s8:  lpx_set_row_name(lp, 3, "r");
s9:  lpx_set_row_bnds(lp, 3, LPX_UP, 0.0, 300.0);

s10: lpx_add_cols(lp, 3);

s11: lpx_set_col_name(lp, 1, "x1");
s12: lpx_set_col_bnds(lp, 1, LPX_LO, 0.0, 0.0);
s13: lpx_set_col_name(lp, 2, "x2");
s14: lpx_set_col_bnds(lp, 2, LPX_LO, 0.0, 0.0);
s15: lpx_set_col_name(lp, 3, "x3");
s16: lpx_set_col_bnds(lp, 3, LPX_LO, 0.0, 0.0);

s17: rn[1] = 1, cn[1] = 1, a[1] =  1.0;
s18: rn[2] = 1, cn[2] = 2, a[2] =  1.0;
s19: rn[3] = 1, cn[3] = 3, a[3] =  0.0;
s20: rn[4] = 2, cn[4] = 1, a[4] = 10.0;
s21: rn[5] = 3, cn[5] = 1, a[5] =  2.0;
s22: rn[6] = 2, cn[6] = 2, a[6] =  4.0;
```

```
s23:  rn[7] = 3, cn[7] = 2, a[7] =  2.0;
s24:  rn[8] = 2, cn[8] = 3, a[8] =  5.0;
s25:  rn[9] = 3, cn[9] = 3, a[9] =  6.0;
s26:  lpx_load_mat3(lp, 9, rn, cn, a);

s27:  lpx_set_obj_dir(lp, LPX_MAX);

s28:  lpx_set_col_coef(lp, 1, 10.0);
s29:  lpx_set_col_coef(lp, 2, 6.0);
s30:  lpx_set_col_coef(lp, 3, 4.0);

s31:  lpx_simplex(lp);

s32:  Z = lpx_get_obj_val(lp);
s33:  lpx_get_col_info(lp, 1, NULL, &x1, NULL);
s34:  lpx_get_col_info(lp, 2, NULL, &x2, NULL);
s35:  lpx_get_col_info(lp, 3, NULL, &x3, NULL);

s36:  printf("\nZ = %g; x1 = %g; x2 = %g; x3 = %g\n", Z, x1, x2, x3);

s37:  lpx_delete_prob(lp);

      return 0;
}
```

หลังจากได้ชุดคำสั่งหลักแล้ว ขั้นตอนต่อไปคือสั่งให้ตัวแปรภาษาซีทำการแปลชุดคำสั่งด้วยคำสั่งต่อไปนี้

```
$gcc  sample.c  –I../glpk-4.0/include ../glpk-4.0/libglpk.a  –o sample.out
```

คำสั่งนี้หมายถึง ให้แปลชุดคำสั่งในไฟล์ที่ชื่อ sample.c โดยใช้ header file ที่อยู่ในแฟ้มที่ชื่อ include และ library file ที่ชื่อ libglpk.a แล้วให้เก็บผลลัพธ์ในไฟล์ที่ชื่อ sample.out ซึ่งเป็นไฟล์ที่สามารถใช้งานได้ และคำสั่งต่อไปนี้เป็นคำสั่งในการเรียกใช้งานไฟล์ sample.out

```
$./sample.out
```

และจะได้ผลลัพธ์คือผลเฉลยของปัญหากำหนดการเชิงเส้น

```
*   0:  objval =   0.000000000e+00   infeas =   0.000000000e+00 (0)
*   2:  objval =   7.333333333e+02   infeas =   0.000000000e+00 (0)
OPTIMAL SOLUTION FOUND


Z = 733.333; x1 = 33.3333; x2 = 66.6667; x3 = 0
```

นั่นคือ ปัญหากำหนดการเชิงเส้นมีผลเฉลยที่เหมาะสมที่สุดคือ

$$z = 733.333$$

ซึ่งเป็นค่าที่ได้จาก $x_1 = 33.3333, x_2 = 66.6667, x_3 = 0$

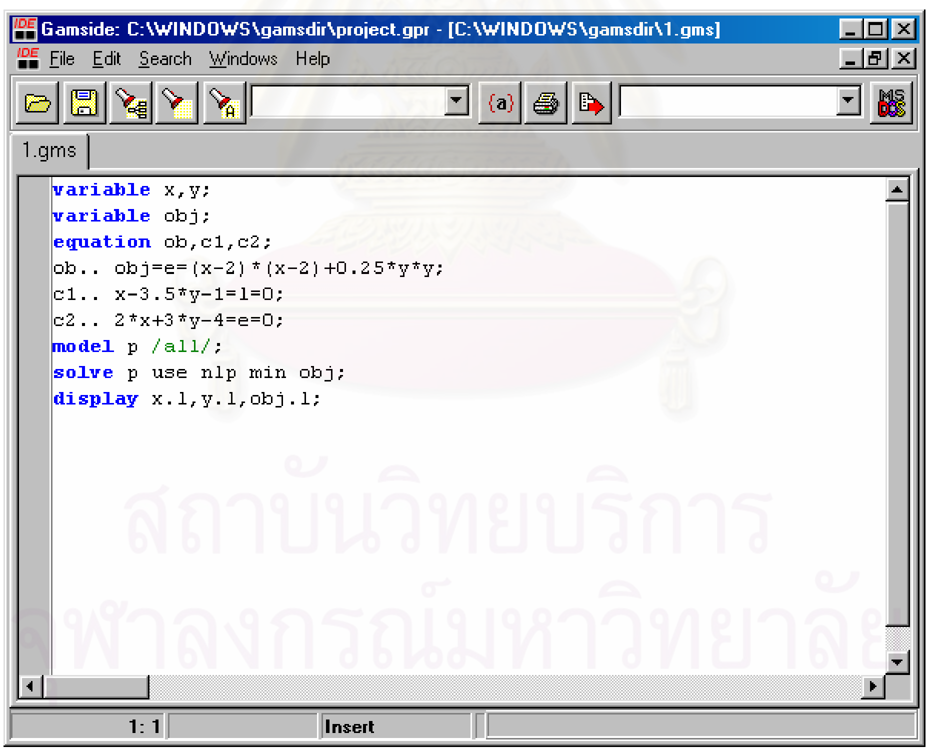
โปรแกรมคำนวณหาผลเฉลยกำหนดการไม่เชิงเส้น GAMS

General Algebraic Modeling System หรือ GAMS เป็นโปรแกรมสำหรับหาผลเฉลยของปัญหากำหนดการเชิงเส้นและปัญหากำหนดการไม่เชิงเส้น พัฒนาขึ้นในช่วงปีค.ศ. 1980-1990โดย GAMS Development Corporation

<u>ตัวอย่าง</u> การใช้งานโปรแกรม GAMS เพื่อหาผลเฉลยของปัญหากำหนดการไม่เชิงเส้น

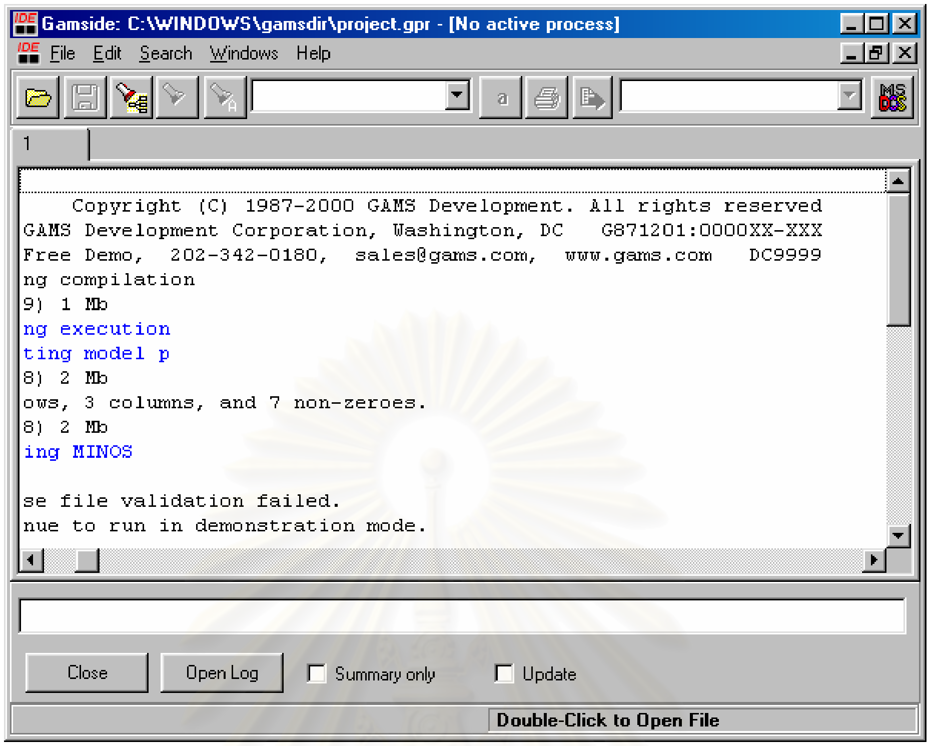
$$
\begin{array}{lll}
\text{mininmize} & (x-2)^2 + \frac{1}{4}y^2 & \\
\text{subject to} & x - \frac{7}{2}y - 1 & \leq 0 \\
 & 2x + 3y - 4 & = 0
\end{array}
$$

จากปัญหาดังกล่าว สามารถเขียนเป็นชุดคำสั่งในโปรแกรม GAMS ได้ดังรูป 6-1 หลังจากนั้นบันทึกไว้ในไฟล์ 1.gms



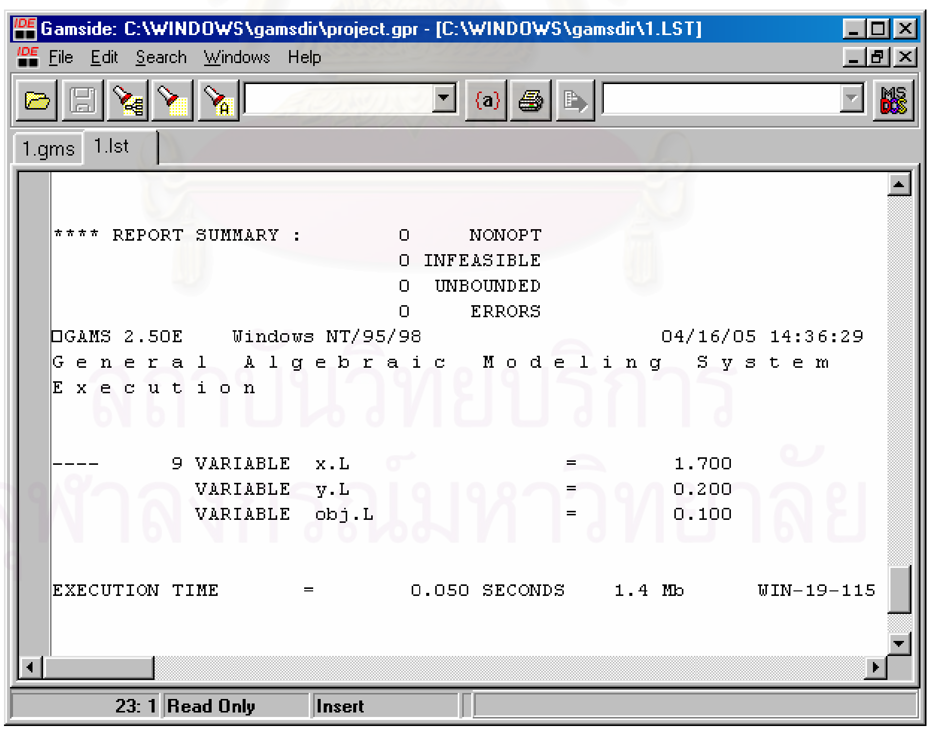
รูปที่ 6-1 แสดงชุดคำสั่งของปัญหาในตัวอย่าง

จากนั้นสั่งรันโปรแกรมจะได้ผลลัพธ์ดังรูป 6-2



รูปที่ 6-2 แสดงผลลัพธ์จากการรันโปรแกรม

หลังจากรันโปรแกรมจะได้ไฟล์ 1.lst ที่แสดงผลลัพธ์ของปัญหากำหนดการไม่เชิงเส้นนี้ ดังรูป 6-3



รูปที่ 6-3 แสดงผลลัพธ์จากไฟล์ 1.lst

จากไฟล์ 1.lst ผลเฉลยเหมาะที่สุดคือ $x = 1.7, y = 0.2$ และค่าฟังก์ชันจุดประสงค์เท่ากับ 0.1

ประวัติผู้เขียนวิทยานิพนธ์

ชื่อ-นามสกุล นางสาวศุภิสรา ศรีขวานทอง

วัน เดือน ปีเกิด 1 พฤษภาคม พ.ศ.2523

ภูมิลำเนา กรุงเทพมหานคร

สำเร็จการศึกษา วิทยาศาสตรบัณฑิต สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ สาขาคณิตศาสตร์ประยุกต์ คณะวิทยาศาสตร์ประยุกต์ ปีการศึกษา 2543