

Prof. Dr. Dr. h.c. mult. Martin Grötschel
Dr. Axel Werner

Programmieraufgabe

Abgabetermin: 12. Januar 2014, 23:59:59 CET

Teilnahmebedingungen. Die Bearbeitung der Programmieraufgaben ist nicht notwendig zum erfolgreichen Abschluss der Vorlesung. Zusammenarbeit in Gruppen beliebiger Größe ist möglich. Die Programme sollen in der Vorlesung am 13. Januar 2014 vorgeführt werden, dabei kann auch der eigene Laptop benutzt werden (sofern er sich an den Beamer anschließen lässt). Die Wahl der Programmiersprache steht frei, sofern sichergestellt ist, dass das Programm unter einem Standard-Linux läuft und sich dort, soweit nötig, auch kompilieren lässt (ein entsprechender Compiler oder Interpreter muss frei verfügbar sein) und sich aus der Kommandozeile (unabhängig von einer Entwicklungsumgebung wie Eclipse) starten lässt.

Vorzuführende Programme bitte bis 12. Januar 2014, 23:59:59 CET an werner@zib.de senden!

Aufgabe 1. Implementieren Sie das Subgradientenverfahren zur Berechnung von unteren Schranken für das symmetrische TSP! Dem Programm soll als Argument der Name einer Datei übergeben werden, die eine zu behandelnde Instanz enthält. Am Ende der Berechnung soll die gefundene untere Schranke auf der Konsole ausgegeben werden, ggf. auch noch weitere Informationen, z. B. Laufzeit, Güte der Schranke o. ä.

Aufgabe 2. Erweitern Sie das Programm aus Aufgabe 1 zu einem Bündelverfahren! Versuchen Sie, geeignete Parameter zu finden, mit denen möglichst gute untere Schranken für möglichst viele Instanzen berechnet werden können.

Aufgabe 3. Implementieren Sie eine oder mehrere primale Heuristiken. Das können z. B. die in der Vorlesung besprochenen sein oder auch selbst erfundene. Bauen Sie Ihre Heuristik(en) als Subroutine zur Berechnung einer oberen Schranke in Ihr Programm aus Aufgabe 1 (oder 2) ein.

Testinstanzen. Eine Menge von Testinstanzen gibt es unter

<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp>

Die Instanzen kommen in unterschiedlichen Formaten, die in der Dokumentation (zu finden unter comopt.ifl.uni-heidelberg.de/software/TSPLIB95/DOC.PS) ausführlich beschrieben sind. Das Programm soll zumindest die Instanzen mit

EDGE_WEIGHT_TYPE: EUC_2D

verarbeiten können; hier sind die Koordinaten aller Knoten in der 2-dimensionalen Ebene gegeben und die Distanzen zwischen zwei Knoten $u = (u_1, u_2)$, $v = (v_1, v_2)$ berechnen sich aus dem Euklidischen Abstand

$$d_{uv} = d_{vu} = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2}.$$

VIEL ERFOLG!