Aufgabe 8.1 - Geschenkeproduktion alias WP

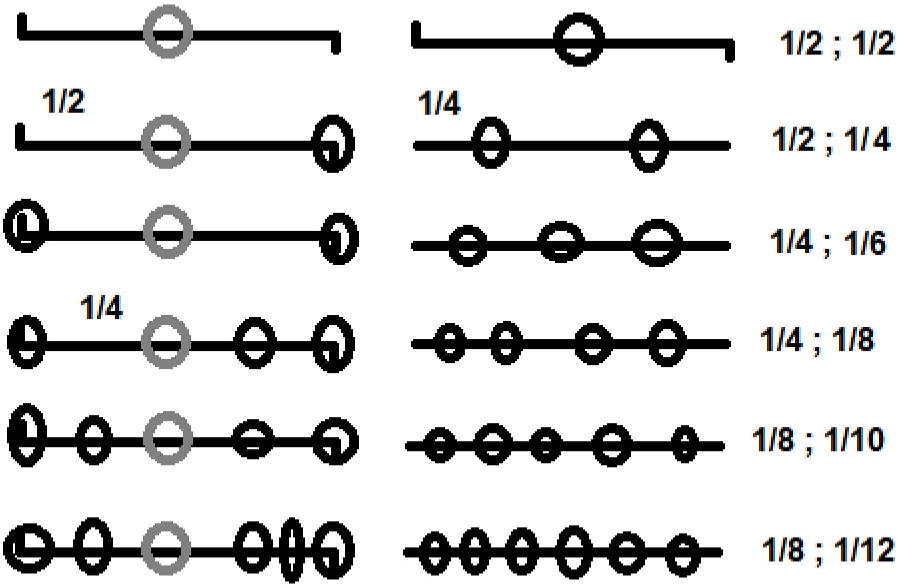
a) G_i als Geschenk, Anfangsbuchstabe für Kind.
G1 : Fussball:
Andrea,Diana,Heidi (A,D,H)
G2 : Buch:
Ben, Chantal, Fabienne (B,C,F)
G3 : Computer:
Chantal, Eve, Georg (C,E,G)
Mit der Spezialisierung der drei Elfen auf Fussball, Buch und Computer lassen sich alle Kinder glücklich machen.
Prüfung: A:G1, B:G2, C:G3, D:G1, E:G3, F:G2, G:G3, H:G1
b)
1. WP liegt in NP:
Wenn wir eine nichtdeterministische Turingmaschine haben, die die Geschenke auswählt, auf die die Elfen sich spezialisieren, dann lässt sich in Polynomialzeit($n \cdot m$) überprüfen, ob mind. ein Wunsch eines jeden Kindes erfüllt wird (erstbester). Vergleiche mit Prüfung bei a), ist für ein Kind kein Geschenk gefunden, so ist es keine Lösung. Die Prüfung erfolgt in Polynomialzeit, damit liegt WP auf jeden Fall in NP.
2. Die Reduktion von Set-Cover:
SC = $\{(A_1, ..., A_n;\text{k})|$es gibt $i_1, ..., i_k$ mit $\bigcup_{j=1}^{k} A_{i_j} = \bigcup_{j=1}^{m} A_j\}$
Also es gibt k Teilmengen, sodass alle Elemente enthalten sind.
Wir wählen die modifizierende Turingmaschine M(w), die zu jedem A_i ein Geschenk G_i bildet und die Elemente dieses A_is als die Kinder angibt, die sich dieses Geschenk wünschen. Diese Umwandlung lässt sich in Polynomialzeit durchführen, in dem für jedes Element aus den A_is ein Kind simuliert wird und das Geschenk G_i in seine Wunschliste eingetragen wird. Falls das Kind noch keinen Wunschzettel hat, dann wird ein neues Kind "kreiert", und in den neuen Wunschzettel das gerade durchlaufene Geschenk eingetragen. Falls das Kind schon existiert, wird nur das Geschenk hinzugefügt.
Formell ist also die Eingabe für Set-Cover $(A_1, ..., A_n;\text{k})$, das k bleibt erhalten. Für jedes A_i wird ein Geschenk angelegt, jedes Element der Gesamtvereinigung wird in ein Kind übersetzt. Wir erhalten hier eine Menge von Mengen von Kindern. Nun wird für jedes Kind eine Menge W_m alias Wunschzettel erzeugt und in diese Menge das Geschenk G_i eingetragen, wenn das K_i in A_i ist. Damit ist die modifizierte Eingabe Der Katalog aller G_i, die Wunschzettel W_m und die Anzahl der Elfen k.
Können nun k Teilmengen gewählt werden, sodass alle Elemente in der Vereinigung enthalten sind, entspricht gerade diese Auswahl den zu wählenden Geschenken, damit jedes Kind mit mindestens einem Geschenk glücklich wird. Denn wenn in der Vereinigung alle Kinder enthalten sind, ist für jedes der Kinder auch mind. ein Geschenk dabei, auf dass es sich freut, nämlich mind. das, bei dem das dem Kind entsprechende Element in die Vereinigung eingefügt wurde. Können k Elfen spezialistert werden, sodass alle Wünsche erfüllt werden, dann sind mit den korrespondierenden A_i auch alle zu den Kindern entsprechenden Elemente enthalten. Falls die Eingabe keine Lösung bietet, können entweder nicht alle Kinder glücklich gemacht werden oder es sind nicht alle Elemente in der gewählten Vereinigung, auch diese Fäll entsprechen sich genau aequivalent.
Damit gilt: $w \in SC \Leftrightarrow M(w) \in WP$.

8.2 - NP vollständige Kaufhauslagereröffnung alias KP

1. KP liegt in NP:
Erhalten wir die Lösung einer nichtdeterministischen Turingmaschine, so lässt sich in Polynomialzeit prüfen, ob alle Geschäfte verbunden sind und die Kosten maximal s betragen. Hierfür müssen die Kosten summiert werden(Linearzeit), für jedes Geschäft geprüft werden, ob es eine Verbindung gibt (Linear in der Anzahl der Geschäfte)
2. Die Reduktion von Vertex-Cover:
VC = {(G;k)|es gibt eine Menge U von k Knoten in G, sodass alle Kanten mind. einen Endpunkt in U haben }
Die modifizierende Turingmaschine M(w), die zu jeder Kante ein Geschäft erstellt und für jeden Knoten ein Lager. Sei m die Anzahl an Geschäften, bzw. Kanten in G, dann wählen wir die Verbindungskosten für je ein Lager A und ein Geschäft B als 1/m, wenn A und B im Graphen inzident sind. Falls sie nicht inzident sind, wählen wir die Verbindungskosten als k+2, wobei k die gesuchte Anzahl aus VC ist. Wir wählen die Eröffnungskosten eines einzelnen Lagers für alle Lager gleich nämlich 1 um diese in Relation zu dem gegebenen k zu setzen. Nun setzen wir die möglichen Gesamtkosten s auf k+1. Damit können nur Lager mit Geschäften verbunden werden, die im Graphen inzident sind, da ansonsten die Kosten höher werden würden.
Gibt es für dieses KP eine Lösung, so sind maximal k Lager gewählt, da jedes Geschäft eine Verbindung benötigt und alle Verbindungskosten sich auf 1 summieren. Es sind also maximal s-1 = k+1-1 = k Lager gewählt, die alle Geschäfte bedienen. Dies entspricht dem Vertex-Cover, da maximal k Knoten gewählt werden aber alle Kanten getroffen sind.
Gibt es eine VC-Lösung, so gibt es maximal k Knoten, sodass jede Kante bedient ist. Die Kosten für k Knoten und den Verbindungen zu n Geschäfte bleibt aber $\leq k+1$ und ist somit eine Lösung für KP.

8.3 - Approximierter Glühwein

Für die optimale Lösung kann die Zeil mit k Ständen so belegt werden, dass der maximale Abstand zur nächsten Bude $\frac{1}{2k}$ beträgt, dies ist mit den äußeren Abstand von $\frac{1}{2k}$ zum Rand und den Buden untereinander mit $\frac{2}{2k}$ gegeben. Das ist stets die Optimale Lösung.
In unserem Mitte-Zuerst Ansatz sind nach den ersten 3 Schritten immer bestimmte Muster erkennbar. Nennen wir die Belegungen, in denen alle Entfernungen gleich sind die vollen Belegungen, so lässt sich zeigen, dass bei k = 2^i +1 immer volle Belegungen durch unseren Ansatz gelingen. Denn beim setzen einer Bude werden zwei neue Teilstrecken erzeugt, die dann wieder halbiert werden. Ich benötoge also immer doppelt soviele Buden wie vorher, um eine volle Beleung zu erzeugen... 3+2 = 5 ; 5+4 = 9 ; 9+8 = 17; 17+16 = 33...., somit immer eine Zweierpotenz plus 1. Damit ist der größte Unterschied immer bei den 2er potenzen: $k = 2^n$. Optimum ist $1/2k$
Unsere Approximation hat dann immer noch ein Wegstück, das noch nicht halbiert ist, damit ist dort die maximale Distanz stets aus der Mitte des noch nicht halbierten Teilstückes: $1/2 \cdot 1/2^{n-1} = 1/k$.
In allen anderen Anzahlen von k ist die Differenz zwischen Approximation und Optimum geringer. Die ersten 6 Wahlen sind in dieser Skizze grob (wirklich nur grob) dargestellt:



Um die maximale Differenz beim minimierungsproblem jetzt mit einem Faktor zu beziffern gilt $F \leq \delta \;\; Optimum$ hier setzen wir die Werte ein, bei denen die größte Differenz herrscht, also eine Zweierpotenz. $\frac{1}{k} \leq \frac{1}{2k}$ und gleichheitherrscht bei $\delta = 2$, alle anderen Werte sind näher dran. Damit ist das Approximationsverfahren 2-approximativ.

8.4 - Approximierte Weihnachtsgäste a) + b)

Jeder mag maximal H Leute nicht. Angenommen es gibt k Personen, die auch genau H Leute nicht mögen. Worst case sie trifft genau die zuerst. Dann reduziert sie jedesmal die maximal mögliche Zahl um H Leute, das kann sie maximal OPT/H mal machen, denn OPT/H * H = OPT. Gilt nun k > OPT/H und dieses mögen auch alle unterschiedliche Personen nicht, dann sind die Überschneidungen leer und sie verbietet so wirklich quasi immer H Menschen. Hat sie dies OPT/H mal gemacht, hat sie im worst case alle anderen ausgegrenzt. Und mehr Leute können nach Vorraussetzung gar nicht eingeladen werden. Somit kommen im schlechtesten Fall eben OPT/H.
Falls Sie jemanden vorher trifft, der weniger Leute nicht mag, oder sich bestimmte unbeliebte Personen überschneiden, kann sie eventuell mehr einladen. Aber OPT/H ist die absolute Untergrenze und wird eben auch im worst case bei den ersten OPT/H Begegnungen mit komplett unterschiedlichen "Nichtfreunden" der Anzahl H erreicht. Somit ist OPT/H ein Minimum und im worst case eben auch ein scharfes Minimum das erreicht wird.