

Sabado 1 de abril de 2017
Universidad Simón Bolívar
Diseño de Algoritmos 1
Erick Silva
Carnet: 11-10969

Informe Proyecto 2: Branch and bound para resolver PRPP

Detalles de la implementación:

Para la implementación se usó la heurística proveída en el enunciado del proyecto, que aunque no era mejor en todos los casos que la implementada en el primer proyecto, se comportaba un poco mejor en CHRISTOFIDES y se beneficiaba más de la eliminación de ciclos negativos (que también fue implementada para este proyecto).

La representación del grafo se hace con matrices de tamaño fijo para el beneficio y costo de cada lado debido a la baja cantidad de vértices en los datos dados, además se usa una lista de adyacencias para mantener el grafo.

Para la implementación del algoritmo de branch and bound como tal se usa una clase auxiliar "lado" que nos permite mantener el beneficio y costo de cada lado utilizado individualmente, en la implementación de branch and bound, a diferencia de la implementación de la heurística, no se edita el grafo y por lo tanto no se requiere que los valores de beneficio sean renovados en cada iteración.

Además de la implementación del algoritmo se implementan dos podas simples adicionales en el mismo para intentar hacerlo un poco más rápido, estas solo tienen que ver con no usar lados con beneficio 0 si no se han usado los originales todavía.

Se encontraron varias dificultades debido a errores en el pseudocódigo (Por ejemplo, en `estaEnSolParcial()` no es suficiente decir que el lado este con beneficio 0, puesto que esto bloquea a los lados que tienen beneficio 0 desde el principio de aparecer dos veces.

Resultados experimentales y análisis:

Las corridas fueron realizadas en una máquina con un procesador intel i5-6400, cuya puntuación en los benchmarks es de 1,830, se asume que este score es lineal, por lo que despejando se obtiene un tiempo de corrida máximo de aproximadamente hora y media, pero debido a la disponibilidad de la computadora y al máximo de poco más de 60 horas necesitadas para estas corridas se decidió usar 30 segundos por instancia.

Debido al uso de windows para realizar las corridas, éstas solo tienen precisión de 3 decimales significativos, por lo que cualquier corrida mas rápida que una centésima de segundo mostrará duración 0.

Una observación curiosa es que hay algunos casos donde branch and bound se ejecuta más rápido que la heurística, esto se debe a que si el backtracking encuentra la solución del problema rápidamente, este puede hacer poda de muchas otras ramas, logrando una muy buena velocidad en los casos que se logra.

El principal aspecto que podemos ver en los resultados es que si bien la duración de cada corrida tiene una relación con el número de nodos, también existe una relación entre la duración y el número de nodos “no beneficiosos” que debe tomar el camino óptimo, esto se debe al ordenamiento que se hizo al elaborar la lista de sucesores, algoritmos de búsqueda más avanzados usan múltiples colas de prioridades con diferentes ordenamientos y heurísticas por esta razón.

Se puede ver que otro aspecto de la duración es probar que una solución es óptima, si bien una solución óptima se puede encontrar en algunos casos fácilmente (Como en G10, cuya solución óptima es no tomar ningún nodo), el demostrar que la misma es óptima requiere una gran cantidad de tiempo.

Además, podemos ver que la duración no depende del beneficio obtenido por el ciclo, esto se puede ver más claramente en las instancias RANDOM, que aunque tienen un gran beneficio muchas corrieron muy rápidamente.

Tablas Resultados.

Instancia	Vo	VHeur	%dHeur	t(seg)
CHRISTOFIDES/P01	3	3	0	0.002
CHRISTOFIDES/P02	66	12	81.818	0.073
CHRISTOFIDES/P03	56	28	50	T
CHRISTOFIDES/P04	45	39	13.333	T
CHRISTOFIDES/P05	35	19	45.714	0.182
CHRISTOFIDES/P06	60	39	35	T
CHRISTOFIDES/P07	89	55	38.202	T
CHRISTOFIDES/P08	90	61	32.222	T
CHRISTOFIDES/P09	46	31	32.609	5.856
CHRISTOFIDES/P10	41	31	24.39	0
CHRISTOFIDES/P11	9	5	44.444	0
CHRISTOFIDES/P12	10	8	20	0
CHRISTOFIDES/P13	5	4	20	0
CHRISTOFIDES/P14	128	76	40.625	T
CHRISTOFIDES/P15	43	3	93.023	2.827
CHRISTOFIDES/P16	113	56	50.442	T
CHRISTOFIDES/P17	42	35	16.667	24.358
CHRISTOFIDES/P18	21	1	95.238	4.89
CHRISTOFIDES/P19	90	31	65.556	T
CHRISTOFIDES/P20	246	184	25.203	T
DEGREE/D0	109	51	53.211	0

DEGREE/D1	115	0	100	0.015
DEGREE/D2	274	92	66.423	0
DEGREE/D3	172	109	36.628	0.027
DEGREE/D4	210	0	100	0.006
DEGREE/D5	313	108	65.495	0.202
DEGREE/D6	166	115	30.723	0.549
DEGREE/D7	260	0	100	0.125
DEGREE/D8	457	356	22.101	1.869
DEGREE/D9	160	0	100	T
DEGREE/D10	0	0	0	3.144
DEGREE/D11	398	0	100	T
DEGREE/D12	280	0	100	T
DEGREE/D13	717	0	100	T
DEGREE/D14	810	335	58.642	T
GRID/G0	0	0	0	0
GRID/G1	0	0	0	0.015
GRID/G2	0	0	0	0
GRID/G3	2	2	0	0.031
GRID/G4	0	0	0	0.39
GRID/G5	4	0	100	0
GRID/G6	9	4	55.556	0.015
GRID/G7	1	0	100	0.201
GRID/G8	4	0	100	0.062
GRID/G9	2	0	100	9.995

GRID/G10	0	0	0	T
GRID/G11	4	0	100	T
GRID/G12	15	0	100	T
GRID/G13	11	1	90.909	T
GRID/G14	14	9	35.714	T
GRID/G15	26	16	38.462	T
RANDOM/R0	1742	0	100	0.005
RANDOM/R1	4253	0	100	3.419
RANDOM/R2	5638	0	100	0.007
RANDOM/R3	18453	3371	81.732	0.253
RANDOM/R4	17316	0	100	0.103
RANDOM/R5	298	0	100	T
RANDOM/R6	12478	0	100	T
RANDOM/R7	9405	5331	43.317	T
RANDOM/R8	14847	0	100	T
RANDOM/R9	17523	0	100	T
RANDOM/R10	17405	0	100	T