

Παρακάτω παρατηρούμε τα εξής αποτελέσματα από τους αλγορίθμους:

Images with local search MIN:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	threshold	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000100.in stance	Incremental (1a)	2	100	10	1	16345194	15101118	0.216542	0.20006	5
euro-night-0000100.in stance	convex hull	2	100	10	1	23110406	22522072	0.306168	0.298374	44
euro-night-0000400.in stance	Incremental (1a)	2	400	10	1	16849010	16648076	0.206847	0.204381	33
euro-night-0000400.in stance	convex hull	2	400	10	1	28885876	28868948	0.354618	0.354411	7080
uniform-0000400-1.in stance	convex hull	2	400	10	1	196196888	192704622	0.356837	0.350486	8276

Images with local search MAX:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	threshold	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000100.in stance	Incremental (1a)	3	100	10	1	57276158	58464406	0.758798	0.77454	7
euro-night-0000100.in stance	convex hull	3	100	10	1	66991396	67185486	0.887506	0.890077	31
euro-night-0000400.in stance	Incremental (1a)	3	400	10	1	59947742	60593586	0.73595	0.743879	67
euro-night-0000400.in stance	convex hull	3	400	10	1	70842394	70842394	0.869699	0.869699	4793
uniform-0000400-1.in stance	convex hull	3	400	10	1	427996102	428768852	0.778427	0.779833	8166

Images with simulated annealing MIN with Local Step:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000100.in stance	Incremental (1a)	1	100	1000	46235538	34998460	0.612531	0.463662	21
euro-night-0000100.in stance	convex hull	1	100	1000	46299120	34685240	0.613373	0.459512	62
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	37079914	32212008	0.455213	0.395452	85
euro-night-0000400.in stance	convex hull	1	400	1000	43372408	39763952	0.532463	0.488163	5764
uniform-0000400-1.in stance	convex hull	1	400	1000	303917800	289585926	0.552757	0.526691	8219

Images with simulated annealing MAX with Local Step:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000100.in stance	Incremental (1a)	1	100	1000	34495320	40358878	0.456996	0.534677	18
euro-night-0000100.in stance	convex hull	1	100	1000	33644612	35870446	0.445726	0.475214	89
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	38023780	42735814	0.4668	0.524648	126
euro-night-0000400.in stance	convex hull	1	400	1000	45944340	50296932	0.564037	0.617472	8128
uniform-0000400-1.in stance	convex hull	1	400	1000	296748120	315313392	0.539717	0.573483	14165
euro-night-0001000.in stance	convex hull	1	1000	1000	48319680	51498974	0.5603	0.597166	416713

Images with simulated annealing MIN with Global Step:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000100.in stance	Incremental (1a)	1	100	1000	38189800	32818932	0.505941	0.434787	3
euro-night-0000100.in stance	convex hull	1	100	1000	41965768	34477078	0.555965	0.456754	37
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	37899992	37547202	0.46528	0.460949	18
euro-night-0000400.in stance	convex hull	1	400	1000	41515648	41453764	0.509668	0.508908	8319
uniform-0000400-1.in stance	convex hull	1	400	1000	317689718	317168582	0.577805	0.576857	7697

Images with simulated annealing MAX with Global Step:

Name File	greedy Algorithm (sort)	Edge Selection	Points	L	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000100.instance	Incremental (1a)	3	100	1000	29861956	37599178	0.395613	0.498116	37
euro-night-0000100.instance	convex hull	3	100	1000	37596998	44874828	0.498087	0.594504	181
euro-night-0000400.instance	Incremental (1a)	3	400	1000	37151518	37403536	0.456092	0.459186	35
euro-night-0000400.instance	convex hull	3	400	1000	45256176	45621918	0.555589	0.560079	11014
uniform-0000400-1.instance	convex hull	3	400	1000	299283554	300379188	0.544328	0.546321	15016
euro-night-00005000.instance	Incremental (1a)	1	5000	1000	47589088	48282488	0.551828	0.559869	438818

Images with simulated annealing MIN with Sub Division:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	M	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	70	29242020	27276500	0.358991	0.334861	90
euro-night-0000400.in stance	convex hull	1	400	1000	70	33627166	32718174	0.412825	0.401666	147
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	100	38738464	35023420	0.475574	0.429966	119
euro-night-0000400.in stance	convex hull	1	400	1000	100	39332004	37154772	0.482861	0.456132	278
uniform-0000400-1.in stance	convex hull	1	400	1000	100	267565454	257880140	0.48664	0.469025	257
euro-night-00005000.i nstance	Incremental (1a)	1	5000	1000	100	137136942	133995086	0.38968	0.380752	1986

Images with **simulated annealing MAX** with Sub Division:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	M	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000400.in stance	Incremental (1a)	1	100	1000	70	42534396	43841900	0.563498	0.58082	22
euro-night-0000400.in stance	convex hull	1	100	1000	70	43008784	45342212	0.569783	0.600696	83
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	100	40604238	43367862	0.498479	0.532407	106
euro-night-0000400.in stance	convex hull	1	400	1000	100	44931656	45403804	0.551605	0.557401	323
uniform-0000400-1.in stance	convex hull	1	400	1000	100	295910690	315244402	0.538194	0.573358	356
euro-night-0005000.in stance	convex hull	1	400	1000	100	42123732	43198184	0.488454	0.500913	849

Παρατηρήσεις:

- Τα αρχεία uniform είναι πιο αργά από τα euronight.
- Ο local search βελτιώνει πάντα το εμβαδό ενώ ο simulated annealing όχι πάντα.
- Όσο πιο μεγάλο είναι το m στο subdivision τόσο καλύτερη βελτίωση επιτυγχάνουμε.
- Τα uniform αρχεία δεν πετυχαίνουν τόσο καλή βελτίωση όσο τα υπόλοιπα.
- Όλοι οι αλγόριθμοι είναι αρκετά γρήγοροι, στον global step βέβαια έχουμε βάλει upper bound στα iterations ίσο με $2 \cdot L$.
- Κάποιες φορές ο greedy αλγόριθμος βρίσκει αρκετά καλό πολύγωνο με αποτέλεσμα ο optimization αλγόριθμος να το κάνει λίγο χειρότερο καθώς του λέμε να κάνει κάποια κίνηση.
- Επειδή υπάρχει σε πολλά σημεία των αλγορίθμων randomization για να αποφύγουμε την εξαντλητική αναζήτηση σε κάθε βήμα, είναι αναμενόμενο να μην βρίσκουμε σε κάθε run την βέλτιστη λύση.

- Όταν χρησιμοποιούμε τον incremental ως greedy αλγόριθμο, το πρόγραμμα τερματίζει πιο γρήγορα.
- Παρατηρούμε ότι οι αλγόριθμοι που ψάχνουν το μαξ εμβαδόν μερικές φορές το πολύγωνο έχει καλύτερο εμβαδόν στην αρχή και όχι μετά από τις βελτιώσεις.